Optimization Methods for Efficient Relay Techniques in Cellular Networks

by

Edgar Arribas Gimeno

A dissertation submitted by in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Telematic Engineering

Universidad Carlos III de Madrid

Advisor:
Vincenzo Mancuso

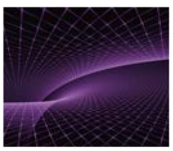
July 2020

Optimization Methods for Efficient Relay Techniques in Cellular Networks

Prepared by:
Edgar Arribas Gimeno, IMDEA Networks Institute, Universidad Carlos III de Madrid
contact: edgar.arribas@imdea.org

Under the advice of:
Vincenzo Mancuso, IMDEA Networks Institute
Telematic Engineering Department, Universidad Carlos III de Madrid

This work has been supported by:



institute IMdea networks

and

This thesis is distributed under license "Creative Commons **Attribution – Non Commercial – Non Derivatives**".

A ma mare, Feli.
A mon pare, Juli.
A la meua germana, Àfrica.

Published Content

This thesis is based on the following published papers:

[1] Edgar Arribas, Vincenzo Mancuso. "Multi-Path D2D Leads to Satisfaction". Published in *IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 12-15 June 2017, Macau, China. http://eprints.networks.imdea.org/1571/1/Multi-Path_D2D_Leads_to_Satisfaction_2017_EN.pdf

- This work is fully included and its content is reported in Chapter 3.
- The author's role in this work is focused on the model design, theoretical analysis, algorithms design and implementation and numerical experimentation with regarding of the concepts proposed in the paper.

[2] Edgar Arribas, Vincenzo Mancuso, Vicent Cholvi. "Fair Cellular Throughput Optimization with the Aid of Coordinated Drones". Published in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 29 April 2019, Paris, France. http://eprints.networks.imdea.org/1966/1/main_Throughput_MiSARN2019_CameraReady_Embedded_CertifiedIEEEeXplore.pdf

- This work is partially included and its content is reported in Chapter 6.
- The author's role in this work is focused on the model design, theoretical analysis, implementation and numerical experimentation with regarding of the concepts proposed in the paper.

[3] Edgar Arribas, Antonio Fernández-Anta, Dariuz Kowalski, Vincenzo Mancuso, Miguel Mosteiro, Joerg Widmer, Prudence WH Wong. "Optimizing mmWave Wireless Backhaul Scheduling". Published in *IEEE Transactions on Mobile Computing*, 2019. http://eprints.networks.imdea.org/2010/1/08745537.pdf

- This work is fully included and its content is reported in Chapter 4.
- The author's role in this work is focused on the model design, theoretical analysis, algorithms design and implementation and numerical experimentation with regarding of the concepts proposed in the paper.

[4] **Edgar Arribas**, Vincenzo Mancuso, Vicent Cholvi. "Coverage Optimization with a Dynamic Network of Drone Relays". *IEEE Transactions on Mobile Computing*, 2019. http://eprints.networks.imdea.org/2027/1/08758183_plusAppendix.pdf

- This work is fully included and its content is reported in Chapter 5.
- The author's role in this work is focused on the model design, theoretical analysis, implementation and numerical experimentation with regarding of the concepts proposed in the paper.

[5] **Edgar Arribas**, Vincenzo Mancuso. "Millimeter-Wave Meets D2D: A Survey". Published in *5G REF Wiley & Sons*, 2020.

- This work is partially included and its content is reported in Chapter 2.
- The author's role in this work is focused on the revision of articles, literature taxonomy analysis and derivation of main proposal findings.

[6] **Edgar Arribas**, Vincenzo Mancuso. "Achieving Per-Flow Satisfaction with Multi-Path D2D". *Ad Hoc Networks*, 2020. http://eprints.networks.imdea.org/2123/1/main_ADHOC_102162.pdf

- This work is fully included and its content is reported in Chapter 3.
- The author's role in this work is focused on the model design, theoretical analysis, algorithms design and implementation and numerical experimentation with regarding of the concepts proposed in the paper.

[7] **Edgar Arribas**, Vincenzo Mancuso. "Fair Throughput Optimization with a Dynamic Network of Drone Relays". *Under revision in Transactions on Networking.*

- This work is partially included and its content is reported in Chapter 6.
- The author's role in this work is focused on the model design, theoretical analysis, implementation and numerical experimentation with regarding of the concepts proposed in the paper.

Abstract

Fast advance in the design of 5G cellular networks has motivated a lot of research that addresses challenges given by the explosive growth of traffic burden, the rise of energy consumption constraints, the unprecedentedly high demand for broadband mobile connectivity and guaranteed quality-of-service for end-users. Therefore the appearance of new technologies, system designs and fast network solutions becomes vital to bear such high demand in network infrastructures.

In this context, the wireless relay scenario has emerged as a key enabler to deal with such challenges. Having clever and efficient schemes that allow traffic to follow alternative relayed paths rather than direct delivery from producer to consumer stands as a crucial need to be properly integrated on the 5G and beyond networks. Depending on the kind of relay, we envision different relay paradigms: users aiming to relief the traffic burden enable device-to-device relay systems; flexible relaying for dense wireless backhaul systems powered by directional transmissions needs smart relay to boost spatial reuse that minimizes the amount of time needed for traffic readiness; and the possibility of mounting relays on extremely-mobile devices such as drones turns the air space into an unexplored vast amount of possibilities to properly position aerial relays.

In this thesis, we present practical optimization tools that leverage the mentioned wireless relay paradigms. We derive optimization frameworks that boost important network metrics such as fair traffic delivery, backhaul traffic readiness or network coverage in current cellular networks. We carefully model network features such as traffic paths, consumed energy, user throughput, transmission directionality or link activation cost, among others. Hence, we approach realistic network infrastructures restricted by technical, physical, flow, or fairness constraints. As unavoidable complex mathematical constraints arise that often turn into an NP-Complete problem, we propose lightweight schemes that work in low-degree polynomial time that are able to provide efficient close-to-optimal solutions, as required in current networks operating at tiny time-scales.

The results reported in this thesis show that designing optimization tools that properly identify key opportunities for efficient relay such as best split traffic paths, best directional transmission scheduling or best aerial relay positioning provides very high gains in terms of throughput experience, fast readiness of traffic at the edge nodes or users coverage. Hence, solutions proposed in this thesis comply with implementation requirements as well as guaranteed performance service for desirable integration on current cellular networks.

Table of Contents

List of Tables

List of Figures

List of Acronyms

3GPP 3rd Generation Partnership Project

5G Fifth Generation

aBS aerial Base Station

AIIS Accumulated Individual Indicator of Satisfaction

AP Access Point

BS Base Station

CDF Cumulative Distribution Function

CSI Channel State Indicator

CSMA/CA Carrier Sense Multiple Access with Collision Avoidance

D2D Device-to-Device

DEMA Dynamic Exponential Moving Average

`DEMM` D2D Expeditious Multi-mode Multi-path

`DIMM` D2D Intensive Multi-mode Multi-path

EIRP Equivalent Isotropically Radiated Power

EMA Exponential Moving Average

eNB evolved Node B

EO *Extremal-Optimization*

EOA *Extremal-Optimization* Algorithm

eProSe Proximity-based Services

FBD2D Floating Band D2D

GAP Generalized Assignment Problem

GMD2D Group D2D Mode

gNB Next Generation Node B

HARQ Hybrid Automatic Repeat Request

HPBW Half-Power Beamwidth

IIS Individual Indicator of Satisfaction

IoT Internet-of-Things

JCR Journal Citation Reports

LTE Long Term Evolution

LoS Line-of-Sight

LP Linear Program

MAC Medium Access Control

μBSs Micro Base Stations

MBS Macro Base Station

MC Monte-Carlo

MCS Modulation and Coding Scheme

MGDC Minimum-Geometric Disk-Cover

MILP Mixed-Integer Linear Program

MINCP Mixed-Integer Non-Convex Program

mmWave Millimeter-Wave

MMWBS mmWave Backhaul Scheduling

MPD2D Multi-Path D2D

MSP Mode Selection Problem

NLoS Non-LoS

`OnDrone` On-demand Drone Coverage

`PADD` Parallelized Alpha-fair Drone Deployment

PEC Partition problem with Equal Cardinality

PPP Poisson Point Process

QoS Quality-of-Service

`RA` Repulsion-Attraction

RAT Radio Access Technology

RF Radio Frequency

RWP Random Way-Point

SCPHY Single-Carrier Physical

SDN Software Defined Network

SDR Software Defined Radio

`Seq Sequential Multi-Placement`

SINR Signal-to-Interference-plus-Noise Ratio

SNR Signal-to-Noise Ratio

SSW Sector Sweep Frame

TDD Time Division Duplex

TDMA Time Division Multiple Access

UE User Equipment

WLAN Wireless Local Area Network

1 Introduction

Cellular networks are experiencing deep changes due to the advent of Fifth Generation (5G) technologies [8]. The need of flexible and adaptive management solutions, to address a highly mutable density of users, has allowed novel communication paradigms to emerge, e.g., Device-to-Device (D2D) communications, as well as smart, flexible and mobile relay, and the use of reconfigurable backhaul links controlled by Software Defined Network (SDN) tools [9]. With precise beamforming and highly efficient cooperative transmission techniques, it is possible to operate broadband wireless backhaul links [10], which are key to promote the use of mobile relays as well as high spatial reuse for novel concurrent relay transmission techniques or high interference-managed resources reuse over several Radio Access Technologies (RATs).

The appealing concept of *D2D Communications* has been proposed in order to improve network performance in sight of the ambitious service optimization goals for network offloading and traffic relay in 5G networks. D2D allows to widen the coverage of cellular networks at lower energy costs through direct links between devices in close range without traversing a Base Station (BS) or the core of the network [11]. D2D-enabled networking has been an important topic of research during the recent years, due to its promising capabilities to accomplish networks requirements. Both cellular (3rd Generation Partnership Project (3GPP)) and Wireless Local Area Network (WLAN) (e.g., WiFi) development organizations have produced specifications for D2D, e.g., the Proximity-based Services (eProSe) in 3GPP [12] and WiFi Direct [13]. Indeed recent research has demonstrated how D2D can be incorporated into cellular networks through WiFi Direct and 3GPP cellular technologies for 5G [14–16].

In addition, one specific RAT that appears as a very interesting technology to address some relay challenges is Millimeter-Wave (mmWave) communication on frequencies from 6 to 300 GHz. The unprecedented vast amount of available spectrum allows for multi-gigabit link speeds and excellent spatial reuse [17], compliant with what expected from D2D performance techniques. mmWave has been proposed for wireless backhauling of small cells. It is particularly well suited for backhauling in extremely dense cell deployments,

where other backhaul technologies are cost intensive [18]. In this context, relay and point-to-point connections as the ones envisioned by D2D communications appear as a key opportunity to address backhaul offloading with parallelized relay streams that reduces the time needed to have the content ready in network edges, as we address.

Finally, the availability of broadband backhaul links allows manned and unmanned vehicles (e.g., drones) to carry mobile relays. The use of mobile relays brings unique opportunities to deploy adaptive and flexible networks that provide connectivity where fixed infrastructures lack operational connectivity [19]. Thus, we also study the case of drones as relay stations transmitting on orthogonal frequencies with respect to ground base stations.

Since the interest for mobile, topology-adaptive and backhaul relays is now reviving—due to the techniques that make it doable in operational networks rather than just in theoretic speculations—in this thesis dissertation we focus on designing effective optimization methods that are able to boost efficient relay techniques in line with the 5G and beyond cellular network paradigm. We combine mathematical optimization and thorough system modelling jointly with specific technical constraints and requirements that make possible the use of efficient relay operation. We integrate several 5G-like RATs (e.g., Long Term Evolution (LTE), WiFi or mmWave), current and future mobility paradigms (e.g., mobile user devices as smartphones and extremely-mobile relays with high processing capabilities as aerial base stations) and newly-emerged control mechanisms relying on D2D communications, 3D-beamforming or highly directional communications. Hence, in this thesis we present a compendium of closely-related relay optimization tools that are key to promote excellent network performance improvement in terms of traffic flow delivery speed-ups, energy efficiency, system fairness and users' satisfaction, coverage, relay mobility and optimal (re-)positioning, aerial relay path planning and high directional and spatial reuse.

1.1. Main Contributions

The main contributions of this thesis have been published in 6 scientific publications from 2017 to 2020. We have published 2 journal papers in the tier-1 journal *IEEE Transactions on Mobiloe Computing* (indexed in Journal Citation Reports (JCR)) and 1 journal paper in the tier-1 journal *Ad Hoc Networks* (indexed in JCR). In addition, another journal paper is currently under review in the tier-1 journal *Transactions on Networking* (indexed in JCR). Regarding conference publications, 1 publication has been published in the *IEEE WoWMoM 2017* conference, Rank A according to CORE2018[1] dataset and Rank A according to ERA2010[2] dataset, and another conference paper has

[1] http://portal.core.edu.au/conf-ranks/
[2] http://www.conferenceranks.com/

been published in the *IEEE INFOCOM WKSHPS 2019 – MiSARN* workshop, collocated with the *IEEE INFOCOM 2019* conference. Finally, we have published one book chapter in the book *5G REF* from *Wiley & Sons*.

More in detail, the contributions of this thesis are the following:

Contribution 1. *Control in Adaptive Multi-Mode D2D Relay Communications in Cellular Networks.*

An adaptive multi-mode D2D optimization framework–Multi-Path D2D (MPD2D)–has been investigated in [1, 6]. MPD2D accounts for the availability of D2D modes under the requirements dictated by a process of flow requests and selects the combination of cellular and D2D links that boost cellular network performance the most, exploiting opportunistic relay. In addition, we formulate a user satisfaction metric that accounts for the history of users within the network. Integrating such a metric is lightweight yet very effective to drive towards almost complete fairness. MPD2D has been developed together with the supervisor of this thesis at IMDEA Networks. This contribution has been partially presented in the *9th IMDEA Networks Annual International Workshop: Enabling Future Internet Applications*, in Leganés, Spain and in the *IEEE WoWMoM 2017* conference, in Macau, China. An extended version of the work appeared in the journal of *Ad Hoc Networks* in 2020.

[1] Edgar Arribas, Vincenzo Mancuso. "Multi-Path D2D Leads to Satisfaction". Published in *IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 12-15 June 2017, Macau, China.

[6] Edgar Arribas, Vincenzo Mancuso. "Achieving Per-Flow Satisfaction with Multi-Path D2D". Published in *Ad Hoc Networks*, 2020.

Contribution 2. *Compact Concurrent Relaying to Optimize mmWave Wireless Backhaul Scheduling.*

An opportunistic scheduling optimization for mmWave backhaul networks is investigated in [3]. In wireless backhaul networks supplied by mmWave technology, we typically find a macro base station that orchestrates the backhaul scheduling of data to be delivered to micro base stations that act as relays for end-users network access. In this context, we investigate an opportunistic scheduling that prioritizes the use of good connections at the macro base station and further leverages compact and concurrent D2D-type relayed transmissions between micro base stations to minimize the time that data needs to be ready at the network fronthaul. In addition, we have complemented this contribution with a survey-type research focused on mmWave-based D2D applications [5]. This research has been developed with the supervisor of this thesis at IMDEA Networks and in collaboration with other researchers from IMDEA Networks, Augusta University (USA), SWPS University of Social Sciences and Humanities in Warsaw (Poland), Pace Uiversity

(USA), and University of Liverpool (UK). This contribution has been also partially presented in the *10th IMDEA Networks Annual International Workshop*, in Leganés, Spain.

[3] Edgar Arribas, Antonio Fernández-Anta, Dariuz Kowalski, Vincenzo Mancuso, Miguel Mosteiro, Joerg Widmer, Prudence WH Wong. "Optimizing mmWave Wireless Backhaul Scheduling". Published in *IEEE Transactions on Mobile Computing*, 2019.

[5] Edgar Arribas, Vincenzo Mancuso. "Millimeter-Wave Meets D2D: A Survey". *5G REF Wiley & Sons*, 2020.

Contribution 3. *Dynamic Multi-Drone Relay Positioning for User Coverage and Fair Throughput Optimization.*
Efficient, dynamic and lightweight methods have been derived to investigate optimal and autonomous repositioning of aerial relay stations over time in order to optimize user coverage performance in [4] and fair system throughput performance in [2, 7]. These methods have been derived based on mathematical modelling and optimization frameworks. Finding best-suited 3-D positions for aerial stations results crucial in order to boost network performance aided by drone relays. These dynamic multi-drone positioning methods have been developed with the supervisor of this thesis at IMDEA Networks and in collaboration with a researcher from Universitat Jaume I (UJI) de Castelló. This contribution has been also partially presented informally at the *INW 2018* workshop, in Courmayeur, Italy and appeared preliminarly in the proceedings of the *IEEE INFOCOM WKSHPS 2019 - MiSARN* in Paris, France.

[2] Edgar Arribas, Vincenzo Mancuso, Vicent Cholvi. "Fair Cellular Throughput Optimization with the Aid of Coordinated Drones". Published in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 29 April 2019, Paris, France.

[4] Edgar Arribas, Vincenzo Mancuso, Vicent Cholvi. "Coverage Optimization with a Dynamic Network of Drone Relays". *IEEE Transactions on Mobile Computing*, 2019.

[7] Edgar Arribas, Vincenzo Mancuso, Vicent Cholvi. "Fair Throughput Optimization with a Dynamic Network of Drone Relays". *Under revision in Transactions on Networking.*

1.2. Outline of the Thesis

The rest of the thesis is organized as follows. First, in Chapter 2 we provide background and related work on D2D communications over several RATs, such as

LTE, WiFi or mmWave, also in accordance with aerial relay-assisted networks. Then, we organize the novel technical content of this thesis in two different parts. Each part discusses a specific scenario of relay-based wireless networks: static and dynamic optimization. Each part presents different chapters that provide details of the contributions mentioned in the previous subsection.

In Part I we propose network optimization solutions that apply to relays for which we cannot influence on their position. Hence, we present novel methods for static relays, such as D2D networks and dense wireless backhauls. In Chapter 3 we present MPD2D, an optimization framework that leverages on multi-mode D2D communications in order to boost cellular network performance. MPD2D splits traffic flows over multiple D2D relay paths and RATs while targeting the optimization of network capacity and energy efficiency. Such research is complemented with the introduction of a newly designed and lightweight user satisfaction metric that accounts for past users' connectivity opportunities and weights future resources allocation to equilibrate users experience over time. In Chapter 4 we approach the problem of integrating high spatial reuse by means of directional relay in mmWave backhaul networks composed by several dense small cells. In that chapter, we analytically study all theoretical implications and understand the challenges that such a complex problem exposes. Furthermore, we propose practical ways of addressing the relay problem in mmWave backhaul networks by means of alternative approximation solutions and studying their performance in current networks.

In Part II we propose an additional extremely mobile paradigm for relayed communications. In this part, we study use-cases in which cellular traffic is relayed to users by means of aerial stations. Hence, we can flexibly manage dynamic repositioning of relays in the air space to boost important network metrics. In Chapter 5, we propose an optimization framework that maximizes coverage, while in Chapter 6 we focus on maximizing the fair distribution of throughput among users. In both cases we present novel optimization frameworks that are able to set up an autonomous network of drones that dynamically reposition to meet target network requirements. Finally, Chapter 7 concludes this thesis.

1.3. Funding

This thesis has been partially supported by the FPU grant (*Ayudas para la Formación de Profesorado Universitario*) from the Spanish Ministry of Education, Culture and Sports (MECD). Grant reference: FPU2015/02051.

2 Background and State-of-the-Art

In this chapter, we introduce background knowledge on current relay solutions, as well as a discussion on the state of the literature related with the topics investigated in this thesis.

Since we study novel relay techniques based on D2D communications, we first introduce this topic in Section 2.1. We discuss what D2D consists in and how different ways of accessing the wireless spectrum generate several alternatives to exploit static relay among user devices. We show the current literature that proposes solutions on this direction and discuss that novel relay techniques proposed in Chapter 3 are needed to further and substantially improve these solutions. Since we also consider in this thesis further short-range outband relay supported by mmWave, we adapt mmWave-based D2D relay solutions to mmWave-powered wireless backhauls. Hence, in Section 2.2 we discuss what kind of mmWave relays have been already investigated and what the related literature has proposed in order to integrate concurrent mmWave-based relay on dense backhaul networks. The outstanding discussed properties of mmWave to be applied on such systems let us investigate in Chapter 4 new relay techniques that schedule boosted backhaul transmissions.

The proposal in last years of developing extremely dynamic networks that rely on drone relays has motivated further research on this thesis to move from the discussed static relay techniques to the possibility of managing mobile relays moving in the air space. Hence, in Section 2.3 we present an open discussion on several non-terrestrial relay alternatives and conclude that drone relays are an excellent feasible option for upcoming years in current cellular networks. A review on aerial relay positioning methods shows that dynamic optimization tools that find best-fitted locations for these relays to optimize important metrics such as coverage or fair capacity is missing in the literature. Hence, in Chapters 5 and 6 we propose novel solutions on this direction.

2.1. Opportunistic Relay through D2D Communications

Coupling D2D technology and cellular networks is not restricted to any wireless technology [11]. A D2D link is a direct connection between two User Equipments (UEs) without traversing a BS or the core of the network, thus providing infrastructureless communications.

2.1.1. D2D Connection Modes

There are different main *D2D modes* proposed to set D2D links between UEs. A *D2D mode* specifies through which band devices connect (licensed or unlicensed) and how UEs access the physical resources. Finding the way of setting the proper D2D mode is key in order to optimize a D2D-network under delay, throughput or energy constraints [20]. The 3GPP provides on technical reports the technological specifications and needs of *eProSe* for enabling D2D communications on current cellular networks [12]. 3GPP enables users to offload and relay traffic, share content, and ensure the D2D network by means of cooperative use of the different D2D connectivity modes.

We present the main D2D modes developed for a D2D-enabled cellular network in Figure 2.1. The frequency spectrum and time are slotted into resource blocks. Depending on which resource blocks devices are assigned to, and how they reuse such resources, different D2D connection modes arise. We also incorporate the *cellular mode* since it is the most common connection mode in a cell.

- Mode 0: **Cellular mode.** Cellular connection between a UE and the BS in either uplink or downlink.
- Mode 1: **Inband Underlay.** D2D connection between two UEs on the licensed band. UEs reuse cellular resources. Spectrum is shared with cellular devices (see red-colored resources in Figure 2.1).
- Mode 2: **Inband Overlay.** D2D connection between two UEs on the licensed spectrum over a resource portion dedicated only to this D2D mode. Spectrum is reused only among D2D UEs over this mode (see green-colored resources in Figure 2.1).
- Mode 3: **Outband.** D2D connection between two UEs in unlicensed spectrum. Medium access control and interference is not under the control of the BS (see blue-colored resources in Figure 2.1).

These modes have been widely studied in literature. Researchers have studied crucial matters in current networks, such as spectral efficiency, power efficiency, Quality-of-Service (QoS) [21], cellular coverage [22], network offloading [23], etc. Each mode enjoys technological properties and offers clearly distinct medium access opportunities, which originates a discussion about which D2D mode fits better in a cellular network [20]. Deciding on which D2D mode/s must be enabled is known in the literature as the Mode

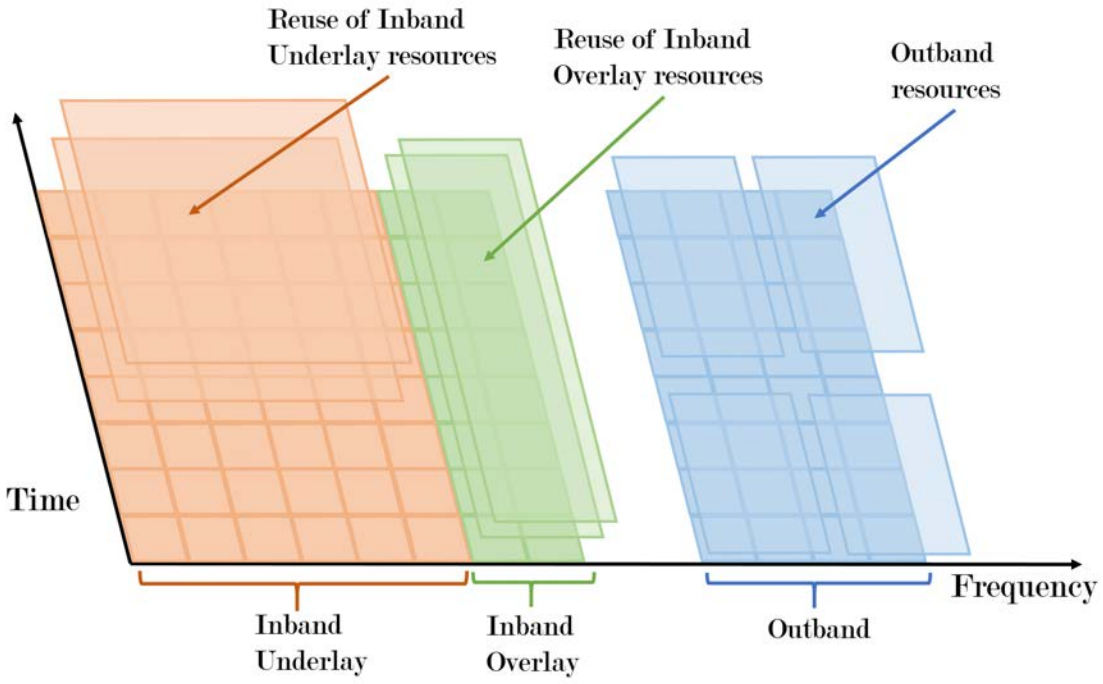


Figure 2.1: Resource allocation in Inband and Outband modes.

Selection Problem (MSP). In Table 2.1 we show the main advantages and drawbacks of each D2D mode, which motivates the integration of frameworks as Floating Band D2D (FBD2D) [24] or MPD2D developed in this thesis, in order to exploit their advantages and reduce the effects of their drawbacks. As these frameworks account for switching from LTE transmission modes to WiFi mode and viceversa, it is important to analyze the complexity overhead of supporting multi-mode frameworks. Asadi *et al.* [15, 16] have experimentally computed the cross-platform delay suffered by a packet that is received by the LTE Medium Access Control (MAC) and processed to the WiFi MAC in a real FPGA-based testbed. This time is of the order of very few milliseconds (below 3 ms), which jointly with the low delay incurred due to LTE and WiFi transmissions complies with the suggested 3GPP delay budget of 70 ms for end-to-end packet delivery [25]. As a result, it is reasonable to assume that a UE is able to switch from one mode to another continuously.

2.1.2. Inband D2D

Li *et al.* [26] propose the use of vertex coloring to jointly perform mode selection and resource allocation in underlay D2D networks. Although they do not model the relay of traffic through underlay D2D, the insights of [26] are relevant to us in order to compare our results in Chapter 3 with the MSP where the only D2D mode enabled is *Inband Underlay*. Conversely, Zhang *et al.* [27] propose Group D2D Mode (GMD2D), a D2D framework where the network performs mode selection in the presence of overlay D2D

Table 2.1: Pros and Cons of each D2D mode

	Advantages	Drawbacks
Mode 1: *Underlay*	Rise of spectral efficiency. Same interface as cellular. SINR control is in BS.	Hard interference control. No D2D plus cellular links.
Mode 2: *Overlay*	Same interface as cellular. Spectrum control is in BS. No cellular interference.	Waste of cellular resources. No D2D plus cellular links.
Mode 3: *Outband*	No cellular interference. Own MAC protocol. Concurrent cellular & D2D links.	Two interfaces in UEs. Harder energy management.

communications. In [27], D2D users establish connections either though cellular links or trough D2D overlay links. In the latter case, the dedicated spectrum may be *divided* or *shared* among D2D users, but always orthogonal to cellular channels. The insights of [27] are also relevant to us in order to compare our proposal MPD2D with frameworks where the only D2D mode enabled is *Inband Overlay*.

Wen *et al.* [28] develop a scheme for mode selection over both *Inband D2D* modes, but they focus on maximizing aggregated network throughput with no energy constraints on users, and impose QoS constraints. Maghsudi *et al.* [29] propose a distributed scheme to approach the MSP on both *Inband* modes where the users are the entities making decisions. However, authors need to over-simplify their problem to find such a distributed solution. Khan *et al.* [30] also study the same scenario, although they propose a scheme in which a circular cell is divided in inner and outer disks for cellular and D2D users, respectively. However, users cannot demand for traffic from multiple entities, only from one neighbor or from the BS. Thus, cellular and D2D users sets are disjoint. Finally, Della *et al.* [31] propose a convex optimization program to solve mode allocation in Time Division Duplex (TDD) systems with one D2D pair transmitting under licensed spectrum.

2.1.3. Outband D2D

Datsika *et al.* [32] propose a cross-network architecture of *Outband* D2D cellular networks, where D2D exploits unlicensed spectrum to relay traffic from the core of the network. By building on IEEE 802.11 mechanisms for channel contention, the authors design an efficient cooperative protocol for *Outband* D2D in one cell. Traffic is efficiently delivered to end-users by means of D2D relays. Here, the use of *Outband* mode as the only D2D enabled mode is relevant to us in order to compare our MPD2D to D2D schemes that only consider WiFi Direct when performing D2D communications. Asadi *et al.*

deploy and study in [33] an interesting architecture based on WiFi Direct integrated to 5G networks. Although they do not consider *Inband* D2D modes, they analyze clustering of D2D users to relay cellular traffic through *Outband* mode, and implement it with a Software Defined Radio (SDR) testbed.

2.1.4. All-band D2D: Integrating Inband and Outband D2D Modes

Authors of [24] propose FBD2D, an innovative framework in which the BS adaptively selects which D2D mode each UE should use for global benefit in a cell. Depending on the scenarios and users density, different D2D modes are allocated to links. For instance, co-channel interference of a microcell spoils *underlay* D2D links, while *outband* WiFi D2D performs better due to the contention of the channel with collision avoidance strategies. Also, low D2D density wastes the *overlay* portion, so that the cellular performance is uselessly decreased. Then, a scenario with heterogeneous communication technologies emerges, facing the MSP defined above. However, the assumptions used for designing FBD2D limit the capabilities of the network. First, FBD2D enforces UEs to use only one wireless interface at a time. Second, FBD2D mainly aims to activate links (cellular or D2D) that connect two nodes without constraints on flows, so that some communications will be likely disrupted. Lastly, FBD2D operates in slotted time, so that every T seconds the scheme optimizes network performance by means of activating links in the next time interval. Nevertheless, in a static scenario the allocation remains the same, so there are links much less utilized that suffer unfair treatment from the network.

All these works do not address the *D2D mode selection problem* as comprehensively as we do. In fact, in Chapter 3 we couple *Outband* mode through WiFi technology, exploit the capabilities of a network with multiple D2D modes enabled where UEs can use several interfaces at once, and account for flow demands accomplished through relaying over multiple D2D paths. Our proposal, MPD2D, reveals that more than 25% of gain in terms of capacity, compared to the closest benchmark from the literature, can be achieved. Besides, we also show how to raise satisfaction of users over time thanks to our newly designed *satisfaction metric*.

2.2. Relaying for mmWave Backhaul Scheduling

While relays on sub-6 GHz bands cause and suffer from significant interference due to their omnidirectional transmissions, the directionality of mmWave antennas mitigates interference, especially in backhaul systems [34, 35]. Multiple links can be active simultaneously as long as their beams do not overlap.

2.2.1. Millimeter-Wave based Relay

Relay techniques based on mmWave permit to extend mmWave multi-gigabit connectivity to coverage areas similar to the ones of conventional microwave networks, but with much lower interference. Many works focus on fully mmWave enabled networks, in which cellular and D2D connections are scheduled on the mmWave spectrum [36–47]. A few works also study mmWave relay features focused on legacy infrastructures supported by microwave technologies, as LTE [48–50], or for Internet-of-Things (IoT) in 5G [51, 52].

Mainly, two kinds of mmWave relays are studied in the literature: dedicated *preplanned* relays and *opportunistic* D2D relays. Preplanned relays are fixed and strategically positioned relay nodes, usually power-supplied to provide alternative routes to traffic flows. Opportunistic relays are basically user devices. There exist also hybrid options, in which preplanned and opportunistic mmWave relays co-exist.

2.2.1.1. Preplanned Relay

Lin *et al.* [45] present a stochastic geometry study of multi-hop mmWave-based relay to evaluate its feasibility. They assume that relay nodes are used to avoid blockage, without considering interference, and show that close-to-optimal connectivity is possible. Biswas *et al.* [41] analyze a similar scenario, but with several source nodes and a single destination, and they consider interference with sectored antennas. Their results show that relay-aided transmissions are able to significantly improve the Signal-to-Noise Ratio (SNR) and enhance coverage and transmission capacity. Xie *et al.* [46] study the coverage performance of mmWave-based relay with several distributions of mmWave BSs, users, blockages and preplanned relay nodes used to avoid blockage. They study the SNR in noise-limited use-cases and the Signal-to-Interference-plus-Noise Ratio (SINR) in interference-limited use-cases. Turgut *et al.* [44] analyze the energy efficiency of mmWave-based relay systems. They model two types of users: non-cooperative and cooperative users. Only the latter use relays to avoid Non-LoS (NLoS) links. With stochastic geometry analysis, the authors conclude that directional mmWave antennas enhance energy efficiency. Finally, Niu *et al.* [47] argue that mmWave-based relay enables fog computing. They adopt multi-hop for mobility-aware caching and concurrent transmissions to exploit spatial reuse. With stochastic optimization, they maximize expected cached hits, although they need to resort to a heuristic, due to complexity.

2.2.1.2. Opportunistic Relay

A few works address opportunistic relay analytically. Wu *et al.* [38] leverage two-hop D2D relay and derive closed-form expressions for the downlink coverage probability. They identify optimal BS deployment densities and point out that the correlation in blockages between cellular and D2D users plays an important role. Another work by Wu *et al.* [39]

provides a wider set of closed-form expressions for several event probabilities in D2D relay systems, when D2D can use mmWave or microwave spectrum, depending on which one provides the best channel. The work shows that D2D relay is beneficial for spectral efficiency and for SINR-based coverage. However, using microwave bands provides better spectral efficiency.

Other works focus on optimization. Kim *et al.* [40] minimize the sum-quality of video streams. Flows may follow multi-hop paths that combine preplanned and opportunistic relays. The work neglects interference but involves constraints for devices, relay and flows. Wei *et al.* [43] derive a throughput-optimal relay probing strategy for two-hop cases. This strategy consists in probing relays until the spectral efficiency of flows is above a threshold. The work shows that there is an optimal threshold and illustrates how to compute it. Eventually, Ma *et al.* [36] optimize mmWave-based relay systems with full-duplex relaying, which causes hard-to-cancel loop interference. They minimize the total transmit power of D2D users and maximize system throughput. Their proposal reduces the total transmit power while improving system throughput.

Sim *et. al.* [37] present the first work involving real mmWave-based D2D experiments. They propose symbiosis between mmWave and D2D over an adaption of the 802.11ad MAC procedure built on top of eProSe. Applied to picocells, they test it over a simple mmWave-based testbed consisting of one BS and two D2D relays. The authors prove that mmWave-based D2D for relay purposes is feasible, although still with several limitations on range, interference and mobility.

2.2.2. Backhaul Networks Powered with Millimeter-Wave

In this dissertation, we analyze wireless backhaul networks and speed up file delivery by means of mmWave relaying. Usually, in the literature preplanned relays are strategically positioned to boost mmWave connections and relay. In contrast, in this dissertation we analyze preplanned mmWave relays that correspond to small BSs that relay backhaul traffic to end-users. Hence, BSs acting as relays have preplanned deployment based on end-users coverage needs, instead of optimal mmWave connectivity environments. Hence, leveraging preplanned mmWave relays, we study opportunistic and compact relaying in mmWave backhaul networks.

The use of mmWave for backhauling small cells in a dense cellular environment enables cost-effective and flexible replacement of the expensive and time-consuming deployment of fiber for gateway access. As discussed in [53, 54], the IEEE 802.11ay amendment, which is the successor to IEEE 802.11ad, includes several modifications that make mmWave suitable for wireless backhauling, among other use-cases. IEEE 802.11ay includes new techniques such as channel bonding and aggregation, non-uniform constellations and enhanced beamforming training that enable peak rates of tens of gigabits per second and allow to build high-speed wireless backhaul networks.

Authors in [26] introduce mmWave backhaul for heterogeneous networks, in which they use joint scheduling and resource allocation schemes based on spatial-division multiple access. They maximize the flow throughput while selecting which paths the content should follow from the BS to the user, but without relaying among Access Points (APs). In [55], the authors also propose a joint transmission scheduling scheme for radio access and wireless backhaul using mmWave D2D communication, where the decision is whether to use a backhaul path or transmit locally among D2D users in case of sufficient proximity. However, although APs relay content through mmWave links, the routes are predetermined by some criterion, instead of minimizing delivery time. Authors in [18] design a mmWave framework for wireless backhaul where flows can follow multiple paths or be served concurrently between two devices. They aim to maximize the aggregated transmission rate, different from what addressed in this dissertation. Finally, Qiao *et al.* [48] envision an mmWave+4G system architecture and propose a Time Division Multiple Access (TDMA)-based MAC for mmWave+LTE based relay supported by reliable 4G signaling. They account for mmWave backhaul (although backhaul relay is not addressed), mmWave access, and under-6 GHz D2D relay and formulate a centralized non-convex mixed-integer maximization of transmitted data. That work concludes that mmWave+LTE based relay in 4G-supported mmWave networks reduces outage probabilities drastically.

Wireless backhaul networks supported with mmWave technology lack of deep studies in the literature that leverage relaying to boost traffic delivery. As discussed above, plenty of work has been done in scheduling communications in related models, including different wireless and optical networks. Nevertheless, link activation cost, interference, and concurrent communication through multiple outgoing links are not taken into account. Even if communication models differ only in minor aspects, problems may be entirely different. Hence, in Chapter 4 we exploit the excellent physical properties of an appealing radio access technology such as mmWave and investigate comprehensive spatial reuse and compact relaying in mmWave backhaul networks under directional interference and actual technological constraints. The optimization tools derived on this direction show that, although it is computationally hard to find optimal or approximation solutions (indeed not possible in polynomial time unless some restrictions are relaxed, as we prove in this thesis), our heuristics and theoretical bounds match network requirements to find a tight transmission scheduling, despite the NP-harness nature of the problem. We show that the delivery time—the makespan in this thesis—of bakchaul traffic can be efficiently minimized by means of simulation results and also real data from experiments.

2.3. Aerial Relaying from Drone Base Stations

The infrastructure of cellular networks is evolving towards flexible and reconfigurable solutions, able to cope with the highly variable densities of users. Specifically, the new generation of cellular communications, namely 5G [8], embeds new transmission techniques as well as novel communication paradigms, including smart and flexible relaying [9]. Besides, wireless relaying with mobility of relays is possible thanks to precise beamforming and highly efficient cooperative transmission techniques, which makes it possible to operate broadband wireless backhaul links [10]. Without such mature technological tools, many attempts toward mobile relaying have failed in the past, since the advent of broadband wireless data networks [56].

It is therefore currently possible to mount mobile relays on, e.g., transport vehicles and drones, which brings the possibility of moving the network with the users and position relays where the fixed infrastructure cannot sustain the user demand [19, 57]. However, there exists serious concerns on the practicality of mobile relaying, due to interference management problems. For instance, Guo and O'Farrel [58] have derived the capacity of OFDMA cellular networks like LTE/LTE-A in the presence of relays reusing cellular frequencies, and showed that relays need to be operated onto orthogonal frequencies. Besides, operating relays over orthogonal frequencies gives additional advantages in terms of simplified resource allocation control [59].

2.3.1. Non-terrestrial Relay Alternatives

The usage of relays operating in the air space through mobile and non-terrestrial devices has been studied for several purposes, over different technologies.

For instance, satellite networks [60] have already been deployed for several years. However, satellites aim to provide service to huge areas, typically at relatively low transmission rates. Moreover, satellites located hundreds of kilometers high are not able to adjust to ground users' topology, neither track the movement of small masses of users, and service incurs high costs. In contrast, drone relay stations may move dynamically at low altitudes and serve smaller target regions on demand, where the ground network cannot sustain the high demand from dense spots, so that a swarm of drones is able to rapidly act for aerial connectivity assistance as the system evolves.

More recently, the Loon and Aquila projects carried by Google [61] and Facebook [62], respectively, have evaluated the operation of aerial base stations mounted on high-altitude platforms (balloons), hovering several kilometers high, and slowly drifting. The Loon project is intended to provide coverage and basic network access to remote and rural areas. Instead, under the aerial relay paradigm, we focus on swarms of small relay stations flying not higher than a few hundred meters, and that can serve broadband links while being easily repositioned on time scales of few tens of seconds.

Drone relays are also different from fixed relays and D2D-based approaches emerged in the last years [63]. In fact, differently from those cases, drone communications are neither fixed nor opportunistic, and the channel propagation is impacted by the probability of communicating with Line-of-Sight (LoS), which varies over time, as we consider in our analysis.

Thus, in general, satellites, balloons or studied terrestrial relays cannot face scenarios as the ones studied under the aerial relay paradigm over a fleet of autonomously coordinated drones. In fact, connectivity requirements used to design protocols for satellite, balloon and D2D communications, as well as technology constraints and signal propagation, are radically different from this case.

2.3.2. Drone Position Optimization

In the recent years there have been various studies that optimize drone relay placement mainly focusing on coverage in static settings and under oversimplified assumptions, as for instance neglecting inter-drone interference [64, 65] or ignoring fairness issues in resource allocation [66]. With that, the resulting problem formulation is simple enough, typically quadratic, yet less realistic and accurate than what we derive in this thesis.

2.3.2.1. Coverage-based Metrics

Al-Hourani *et al.* [67] provide an analytical model for optimal altitude for one drone, to maximize coverage. Also, Hayajneh *et al.* [68] derive optimum drone altitude to minimize outages and bit-error rate. Mozaffari *et al.* [64] study the problem of finding the optimal location for multiple non-interfering drones in order to minimize the total transmission power while satisfying users coverage requirements. The same authors also analyze the performance of a single-drone-aided cell in the presence of underlaid D2D users, by means of stochastic geometry [66] to analyze users coverage. Wang *et al.* [69] use a fleet of drones to optimize aerial optical coverage in which oriented cameras are carried by drones. They build a practical coverage model and test it under simulations and field measurements to get very efficient optical coverage. Strumberg *et al.* [65] propose a moth search algorithm that minimizes the number of drones and optimizes their position in order to monitor a set of ground targets. Chen *et al.* [70] optimize the location of drone relays to provide aerial caching for mobile users that connect to drones by means of millimeter wave links. Petrolo *et al.* [71] have performed real experiments with a machine learning-based system that is able to localize and track mobile users, although they require at least three drones per user. Wang *et al.* [72] optimize aerial drone placement that guarantees coverage of certain users while minimize the required transmit power from drone base stations. They decouple the problem into horizontal and vertical dimensions and solve a basic coverage circle problem. Their approach yields drones positions where altitude and horizontal distance of edge users are proportional.

2.3.2.2. Throughput-based Metrics

Mei *et al.* [73] propose a decentralized inter-cell interference coordination scheme to maximize the weighted sum-rate of one aerial station and all users, as well as the uplink cell association over multiple resources. Guo *et al.* [57] focus on the use of drones as relay stations in cellular networks. Whereas they show that their approach can provide more throughput even in areas of low connectivity, the deployment of the drones does not take into account issues like spectral efficiency. Andryeyev *et al.* [74] estimate drone positions in order to increase cellular capacity by means of a self-organization algorithm based on "repulsion" from base stations and other drones, and "attraction" by mobile users. They use conventional ground path-loss models for wireless channels, which differ substantially from the actual—and more complex—air-to-ground signal propagation model we use in this thesis.

Zeng *et al.* [75] study the use of aerial relays to relay traffic from two ground nodes whose links have been disrupted (due to big obstacles, environment or loss of infrastructures). Authors maximize throughput service and relay trajectories with an efficient algorithm that applies successive convex optimizations. Chen *et al.* [76] extend this problem to the case where the relay network is offered by a swarm of multiple drones, and compare the effects of sending the traffic over one multi-hop link using several two-hop links. Additionally, Zhang *et al.* [77] further focus on the multi-hop link of the aerial network to optimize the trajectories that maximize the end-to-end throughput and minimize transmit powers. These relay problems unveil the potentials of mounting relays on drones, and show clear use-cases for the applicability of such scenario.

Although the related works presented in Subsections 4.3.3 and 4.3.3 that optimize coverage- and throughput-based metrics provide valuable contribution and foundational results, they do not shed light on problems like realistic capacity-based users coverage nor fair capacity maximization (or optimization with fairness targets) when a fleet of drones is deployed to assist a cellular network. In Part II of this thesis, we envision realistic backhaul and backbone constraints in the presence of multiple ground cells that are jointly coordinated with a fleet of many drone relays to realistically target coverage and throughput optimization with fair distribution of resources. The results of this thesis show to considerably outperform several state-of-the-art proposals in terms of both dynamic coverage and capacity, unveil that it is important to account for fairness to benefit from drones that interfere among each other, and show that in some cases unnecessarily large fleets drive a negative impact on the final network performance, among other findings.

2.3.3. Path Planning and Network Architecture

Repositioning of drones and path planning is also of high interest for efficient integration of aerial relaying onto current cellular infrastructures. Fotouhi *et al.* [78] propose a distributed algorithm for autonomous control of drones, and analyze the benefits of repositioning for spectral efficiency using straight paths. However, the literature does not offer yet any clever scheme to design drone paths to assist communication networks. *Bézier curves* [79] have been used to plan drone routes in [80] for military purposes, to smooth drone routes that have to fly over several check-points. However, *Bézier curves* have not been proposed yet to improve communications performance, as we approach for the first time in this thesis.

A complete network architecture that could support a coordinated fleet of drone relays is still under design. Petrolo *et al.* [71] have designed ASTRO, a software-defined network for tetherless coordination of autonomous drones. Sundaresan *et al.* [81] have designed SkyCore, a network module integrated into an end-to-end network architecture called SkyLite. That is a complete network architecture for autonomous drone relay coordination, which demonstrates that operating an aerial network of drone relays is feasible provided the correct optimizations, e.g., coverage maximization (which they do not approach). Hence, as full architectures for future networks have been demonstrated to be feasible, in this thesis we embark on solving optimization problems that find best aerial position for drone relays, provided several physical and technological constraints. Such optimization tools can be run in a module of the network architecture and output the destination positions to which drones will be sent. The maximum coverage framework and the fair capacity system that we discuss in this thesis fit in and are an asset for the above mentioned architectures.

Part I

Static Relay Optimization

In this part, we study relay methods that leverage emerging technologies such as Device-to-Device (D2D)-based networks and new Radio Access Technologies (RATs) such as Millimeter-Wave (mmWave). We envision here relay scenarios from a static point of view, in which features of relays cannot be tuned yet we opportunistically exploit their offered technological possibilities. For instance, user devices dispose of several network interfaces, such as Long Term Evolution (LTE) or WiFi to share their good quality of network connectivity in order to relay traffic and speed up split traffic flows for the network greater good. Also, backhaul wireless networks leveraging mmWave relay use static base stations that mutually coordinate to get traffic ready for delivery to end-users in the minimum time by means of high spatial reuse.

3 Multi-Path D2D: An Optimization Framework for Relayed Traffic Delivery

In this chapter we propose a theoretical framework–*Multi-Path D2D (MPD2D)*–to tackle the *Mode Selection Problem (MSP)* for D2D-enabled cellular networks in a *stateful* manner and under fairness constraints computed not just on individual throughputs but rather on network flows. We exploit both cellular and Wireless Local Area Network (WLAN) technologies to maximize throughput and energy efficiency while targeting flow fairness when *Inband Underlay* and *Overlay*, and *Outband* D2D *modes* are enabled. As main contributions, we develop an optimization scheme that adaptively selects in which mode each User Equipment (UE) will set links to D2D pairs or to the Base Station (BS) in benefit of a network utility function. Unlike past works, we consider flow demands between D2D neighbors and with the BS, and impose their fulfilment through **multi-path** and **relaying**. Indeed, with our proposal, each UE may use simultaneously both cellular and WLAN interfaces for either transmission or reception, thus obtaining higher throughput rates with respect to using only D2D or only cellular connectivity. Moreover, flows can be served over multiple paths within the cellular network, through D2D paths. To reduce complexity and lessen the negative impact on energy consumption and latency due to multi-hop routing, we consider direct cellular links and two-hop D2D paths. Besides, we derive a stateful *satisfaction metric* which is aware of past users' opportunities and increases chances of connections in close future to UEs that enjoyed less throughput.

The design of the satisfaction metric is based on a novel one-step-memory filtering process that we propose, namely the Dynamic Exponential Moving Average (DEMA) scheme. The DEMA scheme measures the satisfaction of users when finite-duration traffic flows are served in the network. By means of integrating proper indicators of users satisfaction and the DEMA scheme, we derive a satisfaction metric that is incorporated to the MPD2D framework. This complete version of the MPD2D framework performs in a much fairer way over time in terms of users satisfaction, while preserving high network performance over time.

In Figure 3.1 we see how flows are split onto heterogeneous multiple paths to get to

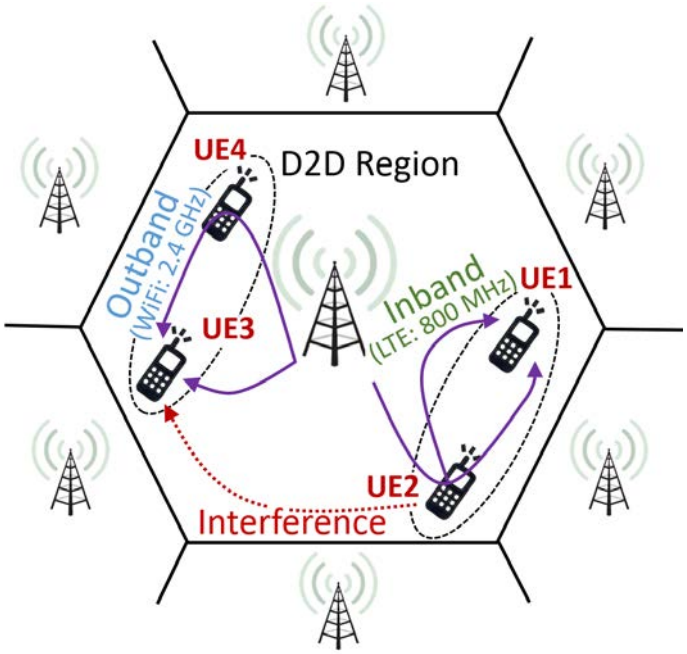


Figure 3.1: D2D-enabled cellular network.

their destinations through relaying and/or direct connections. For instance, the figure shows that there is a traffic flow from the BS targeting *UE1*. This traffic follows first a cellular link to *UE2*. Then, *UE2* relays the traffic over two D2D links exploiting *Inband* and *Outband* modes. Also, the traffic from the BS to *UE3* splits over a direct cellular link and a relayed path through *UE4*. *UE4* downloads part of the traffic for *UE3* and forwards the data over *Outband* D2D mode exploiting WLAN resources. Hence, our proposed framework speeds up traffic flow deliveries reducing the effect of network bottlenecks, by means of multiple traffic paths.

We name our D2D scheme *Multi-Path D2D (MPD2D)*, which results in an NP-hard Binary Non-Linear Program, and propose two effective heuristics in order to approximate the optimal solution of the problem. As a result, we observe that the heuristics for solving MPD2D dramatically increase the performance of cellular networks in comparison to several benchmarks. Specifically, we compare our proposal to the following state of the art solutions/proposals explicitly designed for relay networks based on D2D, namely Floating Band D2D (FBD2D) [24], Group D2D Mode (GMD2D) [27], Underlay MSP [26] and Outband D2D [32].

We summarize the main contributions of this chapter:

- We introduce MPD2D, a multi-mode D2D framework designed to address the MSP in cellular networks in order to optimize network performance in terms of energy consumption and network data flows throughput.
- We propose to use multiple paths to deliver network flows by means of two-hop

D2D relaying with several Medium Access Control (MAC) interfaces.

- We propose the implementation of a satisfaction metric that accounts for past users' experience and increases the connection opportunities in close future for those users that have experienced less benefits from the network in the past. To this end, we derive the DEMA scheme, which has negligible computational cost and achieves very relevant gains in terms of fairness of users' satisfaction.

- We mathematically formulate a Binary Non-Linear optimization program and reformulate it as an Mixed-Integer Linear Program (MILP) in order to solve it with standard tools.

- We propose two effective heuristics, `DIMM` and `DEMM`, that are shown to near optimal, and that reduce complexity and make the deployment of the framework feasible in real cellular networks.

- We perform comprehensive simulations in comparison to state-of-the-art proposals to show the high gains that MPD2D reaches in terms of network throughput, user energy consumption, network efficiency and users' satisfaction fairness over time.

The rest of the chapter is structured as follows. Section 3.1 shows our proposal: MPD2D. Section 3.2 derives the DEMA scheme and builds the satisfaction metric. Section 3.3 presents the optimization formulation. Section 3.4 provides two practical heuristics, while in Section 3.5 we numerically evaluate and benchmark MPD2D and the heuristics. Section 3.6 presents lessons learnt and discussion over this chapter.

3.1. Using D2D Links to Create Multiple Data Paths

In this section we introduce *MPD2D* and show how the use of D2D connections enables multiple data paths for each network flow request. We present the system model and the assumptions we make on D2D-enabled networks. We also model network features and design a network utility function to be maximized based on a trade-off between flows throughput, energy consumption and user satisfaction.

3.1.1. System Model

We consider a hexagonal 3GPP OFDMA D2D-enabled cell with an evolved Node B (eNB) placed in the middle as the BS. The cell is inscribed in a circumference of radius $R_C > 0$ where a set of UE $\mathcal{N}$ are placed (see Figure 3.1). The technology used for *outband* D2D is WiFi. We assume that all UEs have both interfaces, cellular and WiFi, although the scheme can be easily extended to having some UEs with only one interface, or more than two. When two UEs are closer than a D2D pre-defined range $R_{D2D} > 0$ they can establish a D2D connection and become D2D neighbors. Given two UEs in D2D range,

we assign to their link a probability of being D2D neighbors based on distance. Then, let P: $[0,\ R_{D2D}] \to [0,1]$ be a decreasing function, two UEs $u_1, u_2 \in \mathcal{N}$ in D2D range will be D2D neighbors with probability $P(\text{dist}(u_1, u_2))$.[1] We define $\mathcal{N}_u$ as the set of D2D neighbors of UE $u \in \mathcal{N}$ in the cell.

In LTE and in many 5G configurations, downlink and uplink operate separately with a fixed bandwidth. *Inband* D2D modes (either *underlay* or *overlay*) use the uplink bandwidth. For uplink cellular connections the eNB schedules one UE per subframe in a portion reserved for cellular and *underlay* connections. A D2D *inband* link uses all the dedicated bandwidth to either *underlay* or *overlay* mode, so that we manage co-channel interference based on Signal-to-Interference-plus-Noise Ratio (SINR). Hence, cellular UEs do not interfere with each other, but they do with *underlay* UEs. Moreover, D2D users in *overlay* mode do not interfere with cellular nor *underlay* users, but they do interfere among themselves. Additionally, *outband* links will not cause interference since they operate in a different band exploiting WiFi technology. Thus, they contend for the channel with well-known collision avoidance strategies from the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol for IEEE 802.11, as also adopted in [32].

We introduce the presence of cell flows as those entities that need to be provided with a certain Quality-of-Service (QoS). This is important since optimizing per-user throughputs and fairness would not automatically provide per-flow guarantees. When a UE wants to communicate with a device outside the D2D range (either inside or outside the cell) it will do it through the eNB according to the legacy system. Flows originated or terminated outside the cell are mapped onto the eNB as source or destination, respectively. However, if two D2D neighbors wish to communicate, they set a cell flow that must be served through a direct link and/or via the eNB. We define a set of flows $\mathcal{F}$ in which we can find three types of flows:

(i) A cellular flow from the eNB to a destination $d \in \mathcal{N}$.
(ii) A cellular flow from a source $s \in \mathcal{N}$ to the eNB.
(iii) A D2D flow from a source $s \in \mathcal{N}$ to a destination $d \in \mathcal{N}$ in D2D range.

In order to serve such flows, we allow a key feature expected to be performed on D2D communications: **relaying**. Then, a given flow $f \in \mathcal{F}$ may have several paths to follow. We only allow two-hop paths for each flow, and when a path of two hops takes place it must contain the eNB. This approach allows for easy network management at the eNB. Otherwise, in a multi-hop path involving only D2D users, the eNB will hardly be aware of performance regarding interference and signaling in these connections. For instance, WiFi Direct technology has been experimentally proven to be capable of relaying

[1] We assume that the closer two UEs are, the more likely is that they are willing to establish D2D connection.

cellular traffic to users with low cellular access rate [16], supported by cellular signaling with the eNB. However, it poses some challenges for multi-hop D2D paths due to the hard signalling and synchronization management at the eNB, which in contrast to two-hop relayed paths, may incur non-negligible delay and data rate drops [15, 16]. This fact reinforces our choice to consider only two-hop paths in which the eNB is involved. As a matter of fact, if the eNB could have control of all D2D links, irrespectively of whether the traffic went through the eNB, it would be possible to account for multi-hop routing in our framework; however this would imply adding extra and hard-to-manage complexity in the final optimization: while two-hop paths let the optimization account for manageable (and linearizable) quadratic constraints, multi-hop paths imply having polynomial constraints that make the optimization even non-convex, hence not solvable with off-the-self optimizers. In addition, extra constraints accounting for the extra delay are needed, as well as avoiding loops in the paths. Therefore, since the goal of this chapter is to analyze the potentials of relaying in a D2D framework with all the state-of-the-art D2D modes enabled, allowing for relayed paths of two-hops lets us study the gain and opportunities that this multi-mode D2D environment provides to D2D-aided cellular networks.

Clearly, users establishing D2D links for one-hop flows get the immediate benefit of performing direct and fast transmissions at lower power consumption. Additionally, as shown later in Figure 4.a, enabling D2D relay features allows users in the system to experience much more throughput per consumed joule in comparison with non-D2D-enabled schemes. Hence, two-hop relay becomes beneficial at system level. Hence, those users performing D2D communications for relay purposes may be offered bill discounts, and economic benefits in general, in order to incentivize sharing their battery and resources for the sake of system performance. Note also that D2D could be activated within trustable communities (e.g., within the devices of a same group of family members, friends, etc.) where global D2D-enabled communications benefit the community.

We define for each possible flow $f = (s, d) \in \mathcal{F}$ the possible paths that f can follow:

(i) If $f = (eNB, d)$, $d \in \mathcal{N}$, there are two types of possible paths for f:

- A path $\{eNB, d\}$ with downlink cellular mode.
- Any path $\{eNB, r, d\}$ for all $r \in \mathcal{N}_d$.

(ii) If $f = (s, eNB)$, $s \in \mathcal{N}$, there are two types of possible paths for f:

- A path $\{s, eNB\}$ with uplink cellular mode.
- Any path $\{s, r, eNB\}$ for all $r \in \mathcal{N}_s$.

(iii) If $f = (s, d)$, $s, d \in \mathcal{N}$, there are only two paths for f:

- A path $\{s, d\}$ with D2D connection.
- A path $\{s, eNB, d\}$ with uplink and downlink cellular connection where eNB acts as a relay.

We divide the time in intervals of length T seconds. At the beginning of each time interval the eNB first performs *mode selection* and then *resource scheduling*. In the mode selection phase the eNB selects which links will be used according to technology, interference and flow constraints. The eNB selects these links based on the benefits for the overall performance. For resource scheduling we assume the following: (i) cellular resources in mode 0 are allocated proportionally to the number of flows carried by the link; (ii) D2D underlay and overlay (modes 1 and 2) re-use the full uplink bandwidth dedicated to each mode, in each active link. WiFi resources are not scheduled and use instead a classical random access procedure with exponential backoff in case of collision.

3.1.2. Modelling

In this subsection we model all the parameters, variables and metrics in order to analyze the best mode allocation for links, so to maximize the network performance in terms of throughput, energy consumption and satisfaction of users.

Binary decision variables. We assume that at the beginning of each time interval $j \in \mathbb{N}$, the eNB knows the set of cell flows $\mathcal{F}(j)$. Then, the eNB builds a set $\mathcal{L}(j)$ of potentially active links. Denoting by P any of the paths described above, we define:

$$\mathcal{P}_f(j) = \{P \mid \text{flow } f \text{ can follow path } P\},$$
$$\mathcal{L}(j) = \bigcup_{f \in \mathcal{F}(j)} \bigcup_{P \in \mathcal{P}_f(j)} \{(n,m) \in P\},$$

so $\mathcal{P}_f(j)$ is the set of paths that $f \in \mathcal{F}(j)$ can follow. Over the set of links $\mathcal{L}(j)$ we define the decision variables for our scheme and whose values we want to find out in order to optimize the network.

For all nodes $n, m \in \mathcal{N}^*(j)$ such that $(n,m) \in \mathcal{L}(j)$, for all modes $0 \leq i \leq 3$, and for all time intervals $j \in \mathbb{N}$, we define $Y^i_{n,m}(j)$ as:

$$Y^i_{n,m}(j) = \begin{cases} 1, & \text{if } (n,m) \text{ is active in mode } i \text{ during } j; \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathcal{N}^*(j) = \mathcal{N}(j) \cup \{eNB\}$ is the set of nodes of the cell containing the base station. Then, $\{Y^i_{n,m}(j)\}$ is the set of binary decision variables.

Furthermore, to evaluate the energy consumption of WiFi active links, we define an extra set of binary decision variables to tell whether the state of the WiFi interface during time interval $j \in \mathbb{N}$ is idle or the transceiver is used. For mode $i = 3$ and for all

UE $n \in \mathcal{N}(j)$ we define $Y_n^3(j)$ as:

$$Y_n^3(j) = \begin{cases} 1, & \text{if } n \text{ uses the WiFi transceiver during } j; \\ 0, & \text{otherwise.} \end{cases}$$

Throughput modelling. Given a link $(n,m) \in \mathcal{L}(j)$, $\theta_{n,m}^i(j)$ denotes the amount of bits that n would transmit to m in mode $0 \leq i \leq 3$ in time slot j. We model $\theta_{n,m}^i(j)$ as:

$$\theta_{n,m}^i(j) = B_{n,m}^i(j) R_{n,m}^{i,CSI}(j), \qquad 0 \leq i \leq 2; \tag{3.1}$$

$$\theta_{n,m}^3(j) = T \cdot R_{n,m}^{3,CSI} Y_n^3(j-1) + (T - t_{idle}^{act}) R_{n,m}^{3,CSI}(j) \left(1 - Y_n^3(j-1)\right); \tag{3.2}$$

where $B_{n,m}^i(j)$ is the number of resource blocks allocated to link (n,m) and $R_{n,m}^{i,CSI}(j)$ is the number of bits sent per resource block, which depends on the Modulation and Coding Scheme (MCS) and SINR, when $0 \leq i \leq 2$. $R_{n,m}^{3,CSI}(j)$ is the WiFi rate in bps and depends on the number of stations using WiFi. Our throughput model extends what derived in [24] for FBD2D. Unlike previous models, we introduce the time t_{idle}^{act} needed to activate an idle WiFi card, which is the price to pay for the *routing context switch* of (part of) a flow, e.g., to allow for (partial) traffic transfer from cellular to WLAN interfaces in a multi-homed terminal. Hence, in case that a user activates its WiFi card in the current time slot, t_{idle}^{act} accounts for a small pause in service, so that the effective time in which we account for the throughput enjoyed by the user is $T - t_{idle}^{act}$, as shown in Eq. (3.2).

Energy consumption modelling. To model how much energy UEs consume per each mode $0 \leq i \leq 3$ during $j \in \mathbb{N}$, we denote as $E_{n,m}^{i,Tx}(j)$ and $E_{m,n}^{i,Rx}(j)$ the energy spent by $n \in \mathcal{N}(j)$ when she connects to $m \in \mathcal{N}^*(j)$ in mode i during interval j to respectively transmit or receive data. Let $M \in \{Tx, Rx\}$, we model $E_{u_{Tx},u_{Rx}}^{i,M}(j)$ as:

$$E_{u_{Tx},u_{Rx}}^{\kappa,M}(j) = (\beta_{lte} + \beta_{idle}^{WiFi}) \cdot (1 - Y_{u_M}^3(j)) + p_{u_M}^{\kappa,M} t_{B_{u_{Tx},u_{Rx}}^{\kappa}}(j); \tag{3.3}$$

$$E_{u_{Tx},u_{Rx}}^{3,M}(j) = (\beta_{lte} + \beta_{active}^{WiFi}) + p_{u_M}^{3,M} \theta_{n,m}^3(j); \tag{3.4}$$

where $\kappa \in \{0,1,2\}$, β_{lte}, β_{idle}^{WiFi}, and β_{active}^{WiFi} are the baseline energy consumptions in a time interval of length T by LTE, idle WiFi and active WiFi interfaces respectively. For LTE ($\kappa \in \{0,1,2\}$), $p_{u_M}^{\kappa,M}$ ($M \in \{Tx, Rx\}$) is the energy consumed in one subframe for transmission and reception of data, and $t_{B_{u_{Tx},u_{Rx}}^{\kappa}}(j)$ is the number of used subframes. For WiFi ($i = 3$), $p_{u_M}^{3,M}$ is the energy consumed for $M \in \{Tx, Rx\}$ per bit during j. Please note that in the latter case, we account for user energy consumption while the card is being activated (i.e., during the pause in service).

Interference. We build an array $I_{x,m}^i(j)$ that stores the interference that $x \in \mathcal{N}(j)$ causes to $m \in \mathcal{N}^*(j)$ when she transmits in mode i during j. We consider the well-known

path-loss model for wireless transmissions [82]:

$$I^i_{x,m}(j) = p^i_{Tx}(x) \cdot 10^{\frac{-PL(\text{dist}(x,m))}{10}}, \tag{3.5}$$

where $p^i_{Tx}(x)$ is the power of the signal that x used to transmit in mode i, and $PL(\text{dist}(x,m))$ is the path-loss in decibels (dB) which depends on the distance between x and m during j [83].

3.1.3. System Utility Functions

First, we define a node utility function $U_n(j)$ for all users and for the base station. $U_n(j)$ plays a vital role in the definition of the *satisfaction metric* that we will define later. We make $U_n(j)$ account for a trade-off between throughput enjoyed and energy consumed during a slotted time interval j:

$$\begin{aligned} U_n(j) &= \sum_{i=0}^{3} \sum_{m|(n,m)\in\mathcal{L}(j)} \left(\theta^i_{n,m}(j) - \alpha_s E^{i,Tx}_{n,m}(j)\right) \cdot Y^i_{n,m}(j) + \\ &+ \sum_{i=0}^{3} \sum_{m|(m,n)\in\mathcal{L}(j)} \left(\theta^i_{m,n}(j) - \alpha_s E^{i,Rx}_{m,n}(j)\right) \cdot Y^i_{m,n}(j), \end{aligned} \tag{3.6}$$

where θ represents throughput, E represents energy, $\alpha_s > 0$ is a scaling factor for the cost of energy per bit, Y is a binary variable representing the utilization of a link, and the summations extend over network links. Only those links $(n,m) \in \mathcal{L}(j)$ that are active (set to 1) have relevance on the node utility, which is expressed in bits (per interval T). With the above, the global network utility function of the system in a time slot is:

$$U_{net}(j) = \sum_{n\in\mathcal{N}^*(j)} U_n(j), \tag{3.7}$$

where $\mathcal{N}^*(j) = \mathcal{N}(j) \cup \{eNB\}$ is the set of nodes in the cell at time j jointly with the base station itself.

$U_{net}(j)$ accounts for the aggregated throughput and energy consumption of all nodes in the cell. Our aim is to decide link activations in order to maximize $U_{net}(j)$. Then, $U_{net}(j)$ will be the main part of the objective function of our optimization problem, jointly with the satisfaction metric defined next.

3.2. System Satisfaction and the DEMA Scheme.

We introduce satisfaction indicators to measure how users exploit the network over time in order to bias link allocation decisions by following the principles of proportional fair schedulers. When using such schedulers, links acquire priority when they are in good

transmission conditions and if they have been underutilized.

For every time interval $j \in \mathbb{N}$, and for every node $n \in \mathcal{N}^*(j)$, we compute an *Individual Indicator of Satisfaction (IIS)*, namely $f_n(j)$. This indicator tells how good the experience of node n was during interval j. Since the node utility function $U_n(j)$ defined in Eq. (3.6) depends on the decisions made during j and on throughput experienced and energy consumed, a natural definition for IIS is $f_n(j) := U_n(j)$.

Depending on the history of a user, we derive an *Accumulated Individual Indicator of Satisfaction (AIIS)*, namely $F_n(j)$. This value $F_n(j)$ indicates how good the experience of n has been in the previous $j-1$ time intervals. Then, we filter $\{f_n(k)\}_{k=1}^{j-1}$ to obtain $F_n(j)$ with a weighted average:

$$F_n(j) = \sum_{k=1}^{j-1} w_k(j) f_n(k), \quad \text{where } \sum_{k=1}^{j-1} w_k(j) = 1. \tag{3.8}$$

Weights increase with k in an exponential-shaped form, thus concerning more about the satisfaction enjoyed in recent past. This results in a *one-step-memory* filtering process, which avoids having to store each user's full history. Let $\mu > 1$ be a real number. Using the geometric sum result we define:

$$w_k(j) := \frac{\mu - 1}{1 - \mu^{1-j}} \cdot \mu^{k-j}, \quad \forall 1 \leq k < j, \tag{3.9}$$

so the sum of weights $\{w_k(j)\}_{k=1}^{j-1}$ for a fixed j is 1.

Let $\xi(j) := \frac{\mu^{j-1}-1}{\mu^j-1}$ for all $j \geq 1$. Then, the following recurrence for $F_n(j)$ values holds and enables a one-step memory operation:

$$F_n(j+1) = \xi(j) F_n(j) + (1 - \xi(j)) f_n(j), \quad \forall j \geq 1. \tag{3.10}$$

We name our proposal of a one-step-memory filtering process as *DEMA*, since it results to be a novel extension of the well-known Exponential Moving Average (EMA) [84]. Unlike EMA, we have dynamic coefficients for each IIS that are adapted to the lifespan of a flow in the system. As we show below, thanks to the use of dynamic weights, DEMA reacts very quick to changes in node satisfaction. For convenience, we also name as $w_k^{DEMA}(j)$ the weights and $F_n^{DEMA}(j)$ the AIIS values of the DEMA scheme.

In order to average satisfaction values over time with no bias on new arrivals and short-lived flows, we have derived DEMA with no intention to resemble EMA. Hence, to remark the differences between both schemes and the novelty of DEMA, we provide a comparison discussion between both schemes in 3.2.1.

3.2.1. DEMA vs. EMA

In order to average satisfaction values over time with no bias on new arrivals and short-lived flows, we have derived DEMA with no intention to resemble EMA. However, note that EMA corresponds to the following recurrence:

$$F_n^{EMA}(j+1) = (1-\alpha)F_n^{EMA}(j) + \alpha f_n(j), \quad \alpha \in [0,1], \tag{3.11}$$

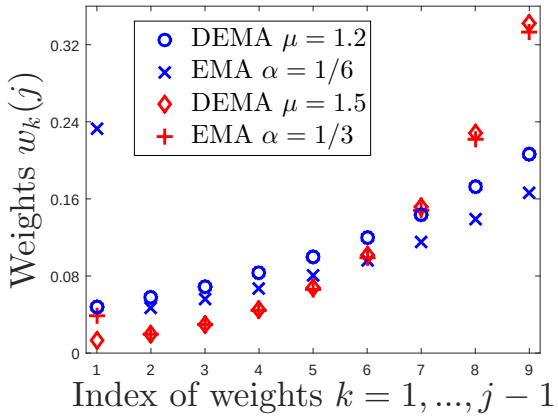
which is similar to the DEMA scheme in Eq. (3.10), although the factor α is constant. Having a constant factor α has a key impact on the behavior of the EMA scheme, and provides substantial differences with DEMA. Indeed, the EMA scheme does not update properly the AIIS values $F_n^{EMA}(j+1)$ because at each new time slot $j+1$, IIS $f_n(j)$ has the same relevance (α), regardless how long the n-th flow has lived in the network. We provide more details in what follows.

Note that using EMA corresponds also to apply a weighted average of the filtered IIS values $\{f_n(k)\}_{k=1}^{j-1}$. The EMA weights, namely $w_k^{EMA}(j)$, are the following:

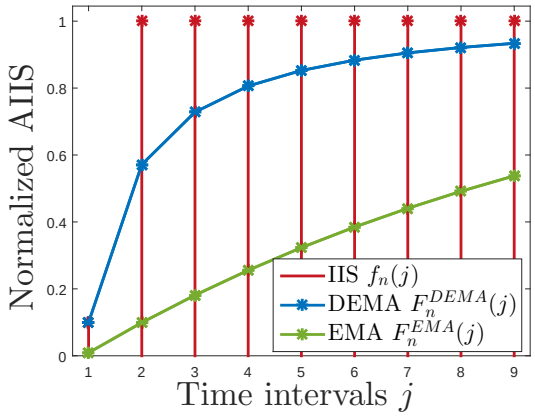
$$\begin{cases} w_1^{EMA}(j) = (1-\alpha)^{j-2}, & \text{if } k=1; \\ w_k^{EMA}(j) = \alpha(1-\alpha)^{j-k-1}, & \forall 2 \le k < j. \end{cases} \tag{3.12}$$

The weights of the EMA scheme do not follow an increasing recurrence, neither are exponentially distributed. Conversely, for DEMA we have conveniently derived increasing exponential-shaped weights (see Eq. (3.9)) that yield a dynamic scheme that adapts to the length of the time history—depending on j—following a one-stem recurrence (see Eq. (3.10)). Although the weights of EMA, $\{w_k^{EMA}(k)\}_{k=2}^{j-1}$, seem to follow also an exponential-shaped curve, the first weight $w_1^{EMA}(j)$ is out of such curve (see Figure 3.2(a)). Such fact has a deep relevance into the performance of the satisfaction metric under the EMA scheme, as we describe below with the help of an example.

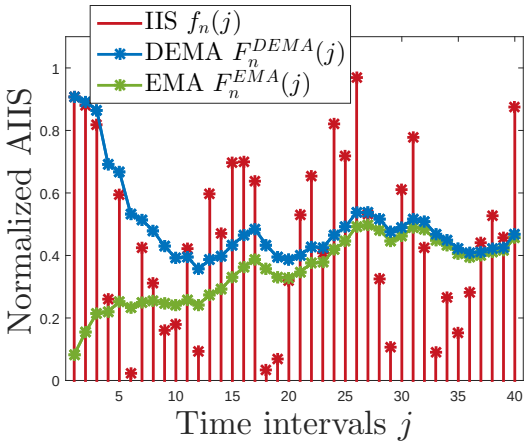
Figure 3.2(a) shows an example of the distribution of the weights of the average filtering of IIS values for both DEMA and EMA using $\mu = 1.2$ and $\mu = 1.5$, and with $\alpha = 1/6$ and $\alpha = 1/3$. While the DEMA weights follow a pure (increasing) exponential curve—since DEMA has been designed with such purpose—we observe that EMA assigns an unnecessarily high value to the first weight, the one corresponding to IIS $f_n(1)$. Such fact goes against the principles of the desired satisfaction metric. The satisfaction metric is intended to gather the aggregate satisfaction of a flow over time by means of assigning much more relevance to the recently past IIS values. Thus, such first weight assignment disrupts the purpose of a satisfaction metric based on EMA. The issue with EMA comes from the fact that the first weight $w_1^{EMA}(j)$ has a different expression with respect to other weights as shown in Eq. (3.12). Specifically, $w_1^{EMA}(j)$ does not include the $\alpha < 1$ factor of the other weights, and so it gives more importance to the oldest time slot.



(a) Weights of DEMA and EMA schemes.



(b) Comparison of DEMA and EMA schemes in short-term history, with $\mu=1.5$, $\alpha=1/3$.



(c) Convergence of DEMA and EMA schemes in long-term history, with $\mu=1.5$, $\alpha=1/3$.

Figure 3.2: Performance of DEMA in comparison with EMA.

Moreover, besides α, the weights in EMA only depend on the distance $j-k$, i.e., the time shift with respect to the current slot. As such, the weights are not adapted to the lifespan of a flow. This would not be an issue with long flows, for which the first EMA weight is practically not biased due to the fact that $\lim_{j\to+\infty} w_1^{EMA}(j) = 0$. However, current cellular networks are extremely dynamic and mobile. Thus, the lifespan of a flow may be short or long—i.e., j may remain small or become large—so distinct flows might experience distinct and variable satisfaction levels during their lifespans in the cell, and thus the satisfaction metric should be flexible enough to account for lifespan. Also, new flows join the cell over time, so that the first period of a flow in the network receives also an unfair treatment from an approach based on EMA. This issue is not present in the design of DEMA, as shown analytically in Eq. (3.9) and illustrated in Figure 3.2(a).

As a consequence, DEMA reacts much better to changes in the individual satisfaction pattern of flows, as shown in Figure 3.2(b). Here, we show a simple yet very illustrative example of the behavior of DEMA in comparison to EMA. We have used an artificial set of IIS values for one flow experiencing an abrupt change of instantaneous satisfaction from 0.1 to 1 at slot $j = 2$. Values used in the example are normalized to one for ease of readability. In Figure 3.2(b), the analyzed flow has received few resources in the first time slot ($j = 1$), so the first IIS value is quite low ($f_n(1) = 0.1$). From the second time slot on ($2 \leq j \leq 9$), the flow finds better signal conditions and receives much more resources, and keeps a constant satisfaction of $f_n(j) = 1$ over the rest of the time. We show the AIIS values $\{F_n(j)\}_{j=2}^{10}$ according to the DEMA (blue in the figure) and the EMA (green) schemes. Here we note that the DEMA scheme immediately reacts to the drastic satisfaction change and increases the accumulated satisfaction $F_n(j)$ over time very fast, according to the IIS values. Thus, DEMA quickly approaches the constant IIS in few time slots. Conversely, the EMA approach shows an undesired behavior in which the AIIS values provided by EMA scheme increase at a much slower rate. Indeed, the EMA approach does not react properly to the drastic IIS change. Observe that after 9 time slots, out of which the last 8 have provided a satisfaction of 1, we expect an AIIS value approaching 1. While DEMA reflects such a behavior, EMA remains approximately 50% far from the desired indicator value.

Finally, in Figure 3.2(c) we show an example similar to the one of Figure 3.2(b). Here, we provide a longer and more dynamic example of (artificial/illustrative) IIS values $\{f_n(j)\}_{j=1}^{40}$ for one flow (again normalized to one). Here, the channel conditions for the analyzed flow are very dynamic, and flow service opportunities vary over time depending on new D2D discoveries, better eNB coverage, or less interference. Again, the DEMA scheme reacts quicker and more efficiently to the high dynamics of the network, while EMA scheme provides from the first time slot satisfaction values that do not adjust to the network dynamism. While the first three flow IIS values ($j \in \{1, 2, 3\}$) are above 0.8 ($f_n(j) > 0.8$), the EMA scheme averages an accumulated satisfaction of $F_n^{EMA}(4) = 0.22$. This fact reveals how undesired and unfair the EMA scheme is in dynamic D2D networks. Here we also show that the AIIS values provided by DEMA and EMA converge to the same value in long-term history, as the time approaches to infinity. This shows again the importance of DEMA scheme for flows with a short lifespan in the network, and for those who have recently joined the network. Thus, while EMA scheme may be appropriate for flows that stay forever in the cell (which is not realistic), the best candidate for highly mobile and dynamic networks is DEMA. We prove such convergence in Lemma 1.

Lemma 1. *Let* $\alpha \in]0, 1[$ *and let* $\mu := \frac{1}{1-\alpha} > 1$. *The asymptotic behavior of EMA and DEMA weights is the same, i.e.,*

$$\lim_{j \to \infty} \left| w_k^{DEMA}(j) - w_k^{EMA}(j) \right| = 0, \quad \forall k \geq 1. \tag{3.13}$$

Proof: Let $j \in \mathbb{N}$ be fixed and let $2 \leq k < j$. Since $\mu = \frac{1}{1-\alpha}$, we have:

$$\begin{aligned} |w_k^{DEMA}(j) - w_k^{EMA}(j)| = \\ \Big|\frac{\frac{1}{1-\alpha} - 1}{1-(1-\alpha)^{j-1}} \cdot (1-\alpha)^{j-k} - \alpha(1-\alpha)^{j-k-1}\Big| = \\ \Big|\frac{\alpha}{1-(1-\alpha)^{j-1}}(1-\alpha)^{j-k-1} - \alpha(1-\alpha)^{j-k-1}\Big| = \\ \alpha(1-\alpha)^{j-k-1}\Big|\frac{1}{1-(1-\alpha)^{j-1}} - 1\Big|. \end{aligned}$$

Since for all $j \in \mathbb{N}$ and for all $2 \leq k < j$ we have that $\alpha(1-\alpha)^{j-k-1}$ is bounded, and it is clear that $\lim_{j\to\infty}\Big|\frac{1}{1-(1-\alpha)^{j-1}} - 1\Big| = 0$, we finally have that:

$$\lim_{j\to\infty} |w_k^{DEMA}(j) - w_k^{EMA}(j)| = 0.$$

Now, let $k = 1$. We have that:

$$\begin{aligned} w_1^{DEMA}(j) &= \frac{\mu - 1}{\mu^{j-1} - 1}; \\ w_1^{EMA}(j) &= (1-\alpha)^{j-1}. \end{aligned}$$

Since $\mu > 1$ and $\alpha \in]0, 1[$, we have that:

$$\lim_{j\to+\infty} w_1^{DEMA}(j) = \lim_{j\to+\infty} w_1^{EMA}(j) = 0.$$

Thus, the claim follows.

■

Lemma 1 shows the relation between the DEMA parameter μ, and the EMA parameter α. Thus, in order to provide a fair comparison between both schemes, the equation $\mu = \frac{1}{1-\alpha}$ must be satisfied (note that we have properly selected the parameters used in the examples of Figure 3.2 according to Lemma 1).

3.3. Optimization of Flow Allocation over D2D Links and Modes

3.3.1. Objective Function

We maximize the network utility $U_{net}(j)$ in a proportional fair way. Therefore, we include AIIS values $F_n(j)$ in the objective function in order to prioritize users not only according to their instantaneous utility, but also according to their satisfaction history.

Then, the lower $F_n(j)$ is for a user flow n, the higher we make its contribution to the overall performance:

$$z(j) = \sum_{n \in \mathcal{N}^*(j)} \frac{U_n(j)}{F_n(j)}. \tag{3.14}$$

Please note that the objective function is a combination of throughput and energy metrics, weighted by means of AIIS values, and binary decision variables (Y). As we aim to maximize system throughput while minimizing energy consumption, we have combined the throughput achieved in the reference interval (θ) and the energy consumption (E) by means of a scaling factor (α_s), as shown in Eq. (3.6), which expresses the economical value of a bit of data with respect to the cost of energy. Once formulated the optimization program, such an approach on the utility function provides Pareto-optimal solutions, i.e., the found optimal solution is such that throughput cannot be increased without increasing energy consumption and energy consumption cannot be deceased without decreasing throughput [85]. Hence, our formulation provides the optimal solution searched in such a multi-objective optimization problem. Please see Appendix A for further details.

3.3.2. Network Constraints

MPD2D is restricted to some conditions that result in three kinds of constraints for the optimization problem:

Technology constraints. In LTE (and 3GPP cellular 5G networks in general) a node can set a direct link (either cellular or D2D) with only one other node of the cell. Then, a node n can only use one of the LTE modes: mode 0 (celluar), mode 1 (*underlay*) or mode 2 (*overlay*). Independently she can use also mode 3 (*outband*).

Interference constraints. In a D2D-enabled network there is co-channel interference that can spoil network performance. Then, we impose SINR thresholds in the form of optimization constraints.

Let $n \in \mathcal{N}(j)$, $m \in \mathcal{N}^*(j)$, and $0 \leq i \leq 2$, and let $\mathcal{T}_m^i > 0$. We want that the SINR experienced in link (n, m) in mode i is above $\mathcal{T}_m^i$ in order to ensure good QoS in cellular connections. We give values to these thresholds according to the implication they have in the lowest MCS guaranteed to users [86].

Flow constraints. We impose the service of flows in $\mathcal{F}(j)$, defined in Section 3.1.1, through relaying data over multiple D2D paths.

3.3.3. MPD2D Optimization Problem

The resulting MPD2D optimization problem is shown in what follows. For ease of readability we denote $e = eNB$ and omit j-dependence.

$$
\begin{cases}
\max \quad z = \sum_{n \in \mathcal{N}^*} \frac{U_n}{F_n}; \\
\sum_{i=0}^{2} \sum_{m|(n,m)\in\mathcal{L}} Y_{n,m}^{i} \leq 1, \ \forall n \in \mathcal{N}; \\
\sum_{m|(n,m)\in\mathcal{L}} Y_{n,m}^{3} \leq 1, \ \forall n \in \mathcal{N}; \\
\sum_{i=0}^{2} \sum_{n|(n,m)\in\mathcal{L}} Y_{n,m}^{i} \leq 1, \ \forall m \in \mathcal{N}; \\
\sum_{n|(n,m)\in\mathcal{L}} Y_{n,m}^{3} \leq 1, \ \forall m \in \mathcal{N}; \\
\sum_{(t,r)\in\mathcal{L}} Y_{n,e}^{0} Y_{t,r}^{1} I_{t,e}^{1} \leq \gamma_{n,e}^{0}, \ \forall n \in \mathcal{N} \mid (n,e) \in \mathcal{L}; \\
\sum_{i=0}^{1} \sum_{(x,y)\in\mathcal{L}\setminus\{(n,m)\}} Y_{n,m}^{1} Y_{x,y}^{i} I_{x,m}^{i} \leq \gamma_{n,m}^{1}, \ \forall (n,m) \in \mathcal{L}; \\
\sum_{(x,y)\in\mathcal{L}\setminus\{(n,m)\}} Y_{n,m}^{2} Y_{x,y}^{2} I_{x,m}^{2} \leq \gamma_{n,m}^{2}, \ \forall (n,m) \in \mathcal{L}; \\
Y_{n}^{3} = \min\left(1, \sum_{m\,|\,(n,m)\in\mathcal{L}} Y_{n,m}^{3} + \sum_{m\,|\,(m,n)\in\mathcal{L}} Y_{m,n}^{3}\right), \ \forall n \in \mathcal{N}; \\
\sum_{i=1}^{3} Y_{s,d}^{i} + Y_{s,e}^{0} Y_{e,d}^{0} \geq 1, \ \forall (s,d) \in \mathcal{F} \mid s,d \in \mathcal{N}; \\
Y_{e,d}^{0} + \sum_{r\,|\,(r,d)\in\mathcal{L}} \left(Y_{e,r}^{0} \sum_{i=1}^{3} Y_{r,d}^{i}\right) \geq 1, \ \forall d \in \mathcal{N} \mid (e,d) \in \mathcal{F}; \\
Y_{s,e}^{0} + \sum_{r\,|\,(s,r)\in\mathcal{L}} \left(Y_{r,e}^{0} \sum_{i=1}^{3} Y_{s,r}^{i}\right) \geq 1, \ \forall s \in \mathcal{N} \mid (s,e) \in \mathcal{F}.
\end{cases}
\tag{3.15}
$$

The first four constraints model technology constraints. The first two are for transmitters and the next two for receivers. It follows the set of three interference constraints. First, we manage interference from cellular users in uplink with *underlay* D2D users in the eNB. Second, we manage interference between cellular and *underlay* D2D users in each of the D2D users. Third, we manage interference for *overlay* D2D users. The last three constraints force the optimization to serve all flows over at least one path.

The optimization problem is binary and non-linear in the objective function and in the constraints. All non-linearities can be linearized with additional binary variables and linear constraints that increase the complexity of the formulation. Hence, as shown in Appendix B, we turn the non-linear optimization program onto a Mixed-Integer Linear Program (MILP). Therefore, we can apply standard approaches as a combination of interior-point methods [87] with a *Branch&Bound* search [88] in order to solve it.

In order to reduce complexity, we propose two effective heuristics that closely approximate the optimum provided by MPD2D.

3.4. Heuristics: DIMM and DEMM

Solving the MPD2D optimization problem shown in Eq. (3.15) is computationally hard. Hence we propose *D2D Intensive Multi-mode Multi-path (DIMM)* and *D2D Expeditious Multi-mode Multi-path (DEMM)*, two heuristics that perform a sequential search of multiple D2D paths through multi-mode selection. As described in Algorithm 1, DIMM executes a full search of multi-paths checking SINR violation at each decision. Instead, as described in Algorithm 2, DEMM makes preliminary decisions based on link allocations that potentially violate SINR thresholds, so SINR constraints are assumed to be respected and no longer checked in the algorithm. Thus, DIMM performs more accurate mode selection while DEMM has lower complexity, which is desirable for scalable decision making.

Both heuristics first allocate all cellular connections to UEs to serve all flows and then add as many WiFi links among D2D users as possible in order to give flexibility when trying to move and split flows during the algorithm. Then they iterate over the set of D2D links $\mathcal{L}^{D2D}$, i.e., links that do not involve the eNB, and try to allocate sequentially each of the D2D modes to each of those links, provided that all flows are served. Besides, DIMM checks the SINR thresholds in order to provide feasible solutions. Conversely, DEMM a-priori bans any possible allocation that likely spoils any SINR constraint. To this end, we randomly sort potential links for fairness in terms of energy cost distribution across potential relays, which is also convenient since it incurs low complexity overhead. When a link allocation increases utility, DIMM and DEMM activate the link and deactivate other incompatible links, according to MPD2D constraints.

Hence, the main differences between DIMM and DEMM are the following. While DIMM directly iterates to find the best combination of allocation modes to links that maximizes the utility, DEMM performs a previous banning of a set of links that potentially incur too much interference to the system, so that enabling them would not be potentially beneficial for the final performance. Such a banning is performed by means of checking a-priori whether the incurred link interference is higher than a portion of the maximum allowed interference (modelled with the parameter γ). Hence, while DIMM checks at each decision point if the whole set of interference constraints are violated or not, DEMM takes advantage of the a-priori banning and is able to assume that such constraints do not need to be checked at each decision making. Of course, this is an approximation used to considerably save complexity by using DEMM, as detailed in the complexity analysis.

Complexity Analysis. We iterate over all D2D links in $\mathcal{L}$ and calculate up to one utility per D2D mode and per link: $3|\mathcal{L}^{D2D}|$ utilities. The cost of each utility computation is linear with the number of users $|\mathcal{N}|$, since technology constraints do not allow the

Algorithm 1 *DIMM: D2D Instensive Multimode Multi-path*

Input: $\mathcal{N}$, $\mathcal{L}$, $\mathcal{F}$: Sets of users, links and flows.
$\{I^i_{n,m}\}, \{\gamma^i_{n,m}\}$: interference parameters.
Output: $\mathbf{Y} = \{Y^i_{n,m}\}$: Set of decision variables.
Initialize:
$Y^0_{s,e} = Y^0_{e,d} = 1 \; \forall \; (s,e),(e,d) \in \mathcal{F}$
for $(s,d) \in \mathcal{F} \mid s,d \in \mathcal{N}$ **do**
  $Y^0_{s,e} = Y^0_{e,d} = 1$
  if $Y^3_{s,u} = Y^3_{u,d} = 0 \; \forall \; u \in \mathcal{N}$ **then**
    $Y^3_{s,d} = 1$
  end if
end for
$\mathbf{Y^?} = \mathbf{Y}$; $max = z = U_{net}(\mathbf{Y})$.
while $\mathbf{Y_{old}} \neq \mathbf{Y}$ **do**
  $\mathbf{Y_{old}} = \mathbf{Y}$
  for $(n,m) \in \mathcal{L}^{D2D}$ **do**
    for $i \in \{1,2\}$ **do**
      $Y^{i,?}_{n,m} = 1; \; Y^{k,?}_{n,m} = 0 \; \forall \; k \in \{1,2\} - \{i\}$
      $Y^{k,?}_{n,u} = Y^{k,?}_{u,m} = 0 \; \forall \; u \neq n,m; \forall \; 0 \leq k \leq 2$
      $z = U_{net}(\mathbf{Y^?})$
      if $z > max$ & SINR and Flows satisfied **then**
        $\mathbf{Y} = \mathbf{Y^?}$; $max = z$
      else
        $\mathbf{Y^?} = \mathbf{Y}$
      end if
      for i=3 **do**
        $Y^{3,?}_{n,m} = 1; Y^{3,?}_{n,u} = Y^{3,?}_{u,m} = 0 \; \forall \; u \neq n,m$
        $z = U_{net}(\mathbf{Y^?})$
        if $z > max$ & Flows satisfied **then**
          $\mathbf{Y} = \mathbf{Y^?}$; $max = z$
        else
          $\mathbf{Y^?} = \mathbf{Y}$
        end if
      end for
    end for
  end for
end while

Algorithm 2 *DEMM: D2D Expeditious Multimode Multi-path*

Input: $\mathcal{N}$, $\mathcal{L}$, $\mathcal{F}$: Sets of users, links and flows.
$\{I^i_{n,m}\}, \{\gamma^i_{n,m}\}$: interference parameters. $\rho \in]0,1[$.
Output: $\mathbf{Y} = \{Y^i_{n,m}\}$: Set of decision variables.
for $(n,t) \in \mathcal{N} \times \mathcal{N} \mid I^1_{t,e} > \rho \cdot \gamma^0_{n,e}$ **do** $\theta^1_{t,r} = 0, \forall\ r \in \mathcal{N}$
end for
for $(n,m) \in \mathcal{L}^{D2D}$ **do**
  for $x \in \mathcal{N} - \{n\} \mid I^0_{x,m} > \rho \cdot \gamma^1_{n,m}$ **do**
    $\theta^1_{n,m} = 0$
  end for
  if $\theta^1_{n,m} > 0$ **then**
    for $x \in \mathcal{N} - \{n\} \mid I^1_{x,m} > \rho \cdot \gamma^1_{n,m}$ **do**
      $\theta^1_{x,r} = 0, \forall\ r \in \mathcal{N}$
    end for
  end if
end for
for $(n,m) \in \mathcal{L}^{D2D}$ **do**
  for $x \in \mathcal{N} - \{n\} \mid I^2_{x,m} > \rho \cdot \gamma^2_{n,m}$ **do**
    $\theta^2_{x,r} = 0, \forall\ r \in \mathcal{N}$
  end for
end for
Do `DIMM` without SINR checking.

system to have more that $5|\mathcal{N}|$ active links in the cell.[2] Besides, `DIMM` has to check SINR constraints, whose number is linear with $|\mathcal{L}|$. Therefore, the complexity of `DIMM` is $\mathcal{O}(5 \cdot 3|\mathcal{L}^{D2D}||\mathcal{L}||\mathcal{N}|)$, while `DEMM` has complexity $\mathcal{O}(5 \cdot 3|\mathcal{L}^{D2D}||\mathcal{N}|)$. Since $\mathcal{L}^{D2D} \subset \mathcal{L}$ and the sizes of the two sets are at most quadratic with the number of users, the complexity of `DIMM` goes with the fifth power of $|\mathcal{N}|$ while `DEMM` has a cubic dependence on $|\mathcal{N}|$.

3.5. Numerical Evaluation

In this section we present results for solving MPD2D and for its heuristics through numerical simulations. We study the gain of MPD2D and heuristics in comparison to the benchmarks based on D2D schemes that consider only one D2D connection mode, as GMD2D [27] (only *Inband Overlay* is enabled), Underlay MSP [26] (only *Inband Underlay* is enabled) and Outband D2D [32] (only *Outband* is enabled). Also, we compare to the cellular legacy system. Additionally, we compare FBD2D [24] (all D2D modes are enabled). We mainly study performance of network utility, throughput, energy consumption, system efficiency, fairness and evolution of satisfaction over time. Error bars in the graphs represent 95% confidence intervals.

[2]Each node can have one downlink connection from the eNB, two WiFi links to transmit and receive packets to relay using the random access technique of IEEE 802.11, and one or two links using the uplink licensed band: either a connection to the eNB or a pair of incoming/outgoing D2D inband links.

Table 3.1: Evaluation parameters

Parameter	***Value***
–Cell Deployment–	
Cell and D2D Range R_C, R_{D2D}	175 m, 20 m
Carrier Frequency	Band 20: 800 MHz
Cellular BW (UL & DL)	20 MHz
Overlay Portion	0.3
Time interval length T	2 secs
Time Activation WiFi Card t_{idle}^{act}	300 μs
Thermal Noise Power	-174 dBm/Hz
WiFi Rate	60 Mbps
SINR Threshold	15 dB
–Power & Energy Consumption–	
eNB/Cellular Tx Power	44 dBm / 24 dBm
D2D Inband Tx Power	10 dBm
LTE baseline β_{lte}	1288.04 mW
WiFi baseline β_{idle}^{WiFi}, β_{active}^{WiFi}	77.2 mW, 132.86 mW
WiFi power Tx/Rx	460 mW / 440 mW
Relative Cost of Energy α_s	1 bit/Joule

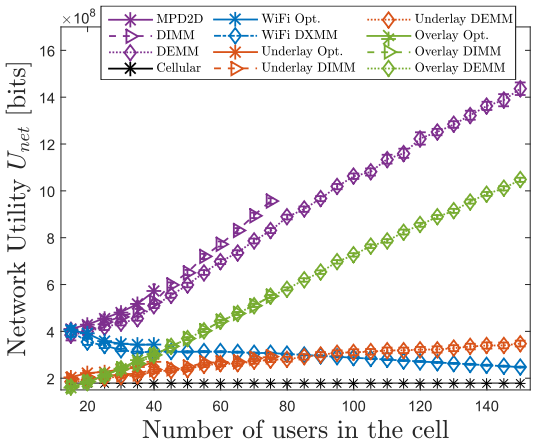
We allocate cellular resources to LTE links proportionally to the number of flows they carry. The amount of D2D users is not limited. Unless otherwise specified, every user has a flow coming from the eNB, and half of them have a flow towards the eNB. We place users uniformly in a hexagonal cell inscribed within a radius $R_C = 175$ m. Any pair of devices becomes a D2D pair with a probability that decreases linearly with their distance and becomes 0 at distances larger than R_{D2D}. Such a D2D range has to be short since it represents a reasonable distance in order to achieve high transmission gains as well as perform communications at high rates (as expected and demanded for D2D to be viable), in any of the modes. Note that long D2D ranges would require unsustainably high transmission power and incur high interference, which would impact the system performance by consuming more energy and adding extra computational cost due to harder interference management. For simplicity, in our numerical experiments, we set $R_{D2D} = 20$ m for all D2D modes, which is in line with what commonly considered in the literature [24, 26, 89, 90]. All D2D pairs may have a flow from one to the other. Any flow can follow multiple paths, as detailed in Section 3.1.1.

We consider as benchmark schemes for MPD2D the following cases: `Overlay`, `Underlay` and `WiFi`. In `Overlay`, the only D2D mode enabled is *Inband Overlay*, as done in GMD2D [27]; in `Underlay`, the only D2D mode enabled is *Inband Underlay*, as done in Underlay MSP [26]; and in `WiFi`, the only D2D mode enabled is *Outband*, as

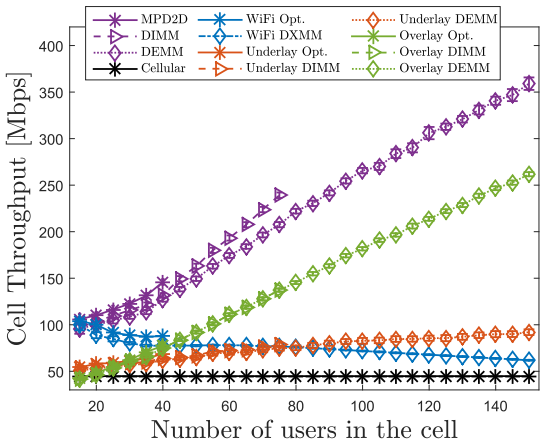
done in [32], by means of WiFi technology. We also compare to `Cellular`, a scheme that corresponds to sending all traffic through cellular connections, as in the cellular legacy system (D2D is not enabled). In `WiFi` scheme, the devices may also use cellular and WLAN interfaces at the same time. We adapt `DIMM` and `DEMM` to work with these three benchmarks, e.g., by disabling the selection of non-allowed D2D modes in each case.

Table 3.1 gathers the main parameters of the system model. We dedicate 30% of cellular resources to *overlay* mode, but leave this portion to cellular and *underlay* modes when *overlay* mode is unused. Carrier frequency for LTE is the 800 MHz band, since it is one of the bands used to deploy 4,5G in European countries (band 20) [91]. Bandwidth for downlink and uplink is of 20 MHz each. The minimum threshold for SINR is 15 dB, so that nodes may use at least a MCS with 16QAM modulation and coding rate of 3/4 [86]. The value of α_s is an estimation of the relative cost of bit with respect to the cost of a Joule in the market, as considered in the literature [24], although other values may be applied. We consider traffic queues under infinite offered load conditions in which users have always data ready to transmit, so we study the achievable performance of the system. We have used `MATLAB R2018a` to implement the MPD2D framework and heuristics so to derive the results. Concretely, in order to find optimal settings, we have used CVX [92,93], a toolbox designed for solving optimization programs that integrates, for instance, interior-point and Branch&Bound methods. We have simulated channel conditions according to the path-loss model used in Eq. (3.5). User positions have been simulated according to uniform distribution within a hexagonal cell inscribed in a circumference of radius $R_C = 175$ m. D2D associations have been set according to the analyzed scenario (D2D pairs need to be always within D2D range, but pairing depends on the scenario. For instance, we mainly test when pairing depends on a probability function than decreases linearly with the distance, as detailed later. Also, we test that users in range are always paired, et cetera). Moreover, when we test the performance of the framework over time, we simulate that users move at a speed of 4 km/h according to the well-known random way-point model, updated every $T = 2$ s, in order to re-optimize the network. We have simulated each scenario 1000 times in order to gather average results in a personal computer.

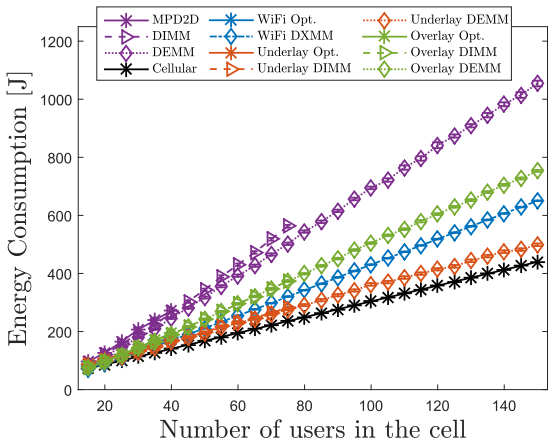
Optimality, Throughput and Energy Consumption. In Figure 3.3 we show a time interval snapshot of the performance of the network utility, cell throughput and user energy consumption. We show optimum values from 15 to 40 users. Both `DIMM` and `DEMM` heuristics provide close approximations for MPD2D and for benchmarks, while allowing to evaluate performance under a larger range of users (up to 150 in the reported figures). As expected, `DIMM` performs closer to optimum values, but it has higher complexity. Then we focus on `DEMM` for heuristic results, since it offers good approximations at quite lower cost. MPD2D clearly outperforms any other case with one single D2D connectivity path enabled. The gain of the network utility compared with `Cellular` with $|\mathcal{N}| = 40$ users is of 218%, while for the densest scenario ($|\mathcal{N}| = 150$) with `DEMM` rises up to 701%.



(a) Network Utility U_{net}.



(b) Aggregated cell throughput.

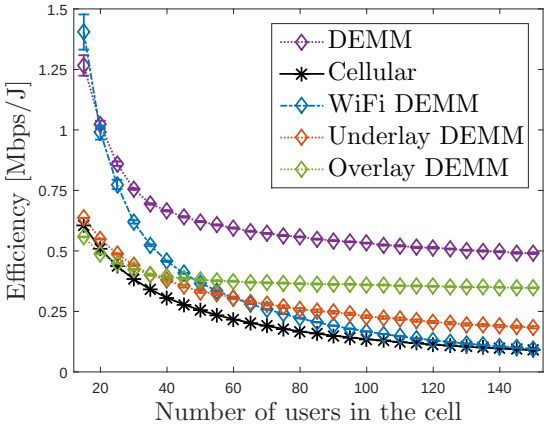


(c) Aggregated energy consumption.

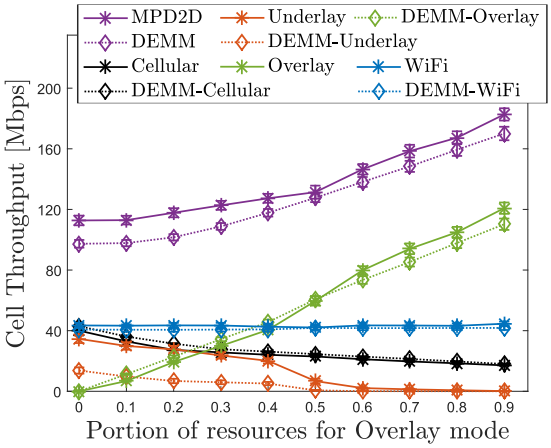
Figure 3.3: Impact of users density on system optimality, throughput and energy consumption.

Furthermore, `DEMM` gives a gain of 37% in comparison to `Overlay`, which offers the best results among the benchmark schemes. In conclusion, as long as we add users, `DEMM` results approximating the optimal MPD2D solution and enjoys much more gain than any other benchmark.

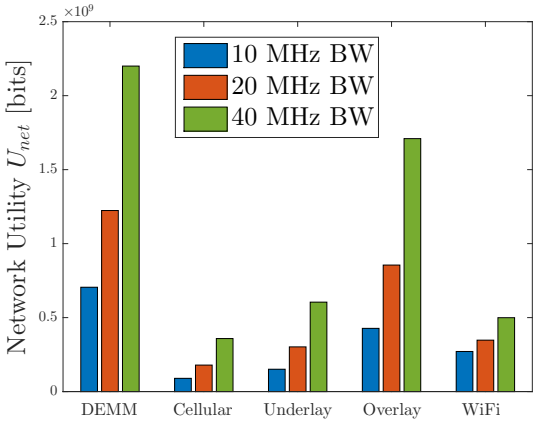
In Figures 3.3(b) and 3.3(c) we can see the specific gain of throughput and energy consumption. The shape of the throughput graph is similar to network utility since throughput is the main part of the utility function in the scheme. Throughput gain grows from 3.6% with $|\mathcal{N}| = 15$ up to 66% with $|\mathcal{N}| = 40$ in comparison with the closest benchmark, `WiFi`. With $|\mathcal{N}| = 150$, `DEMM` provides a gain of 36% in respect to `Overlay`, which for dense cells gives better throughput due to resource reuse, while WiFi usage decreases due to contention for the channel. In comparison to `Cellular`,



(a) Impact of users density on the ratio between throughput and energy.



(b) Impact of *overlay* portion with $|\mathcal{N}| = 30$.



(c) Impact of LTE channel bandwidth with $|\mathcal{N}| = 120$ on all D2D schemes.

Figure 3.4: Network efficiency, impact of overlay portion and impact of LTE channel bandwidth.

throughput gain rises from 135% up to 702%. Besides, regarding energy consumption `DEMM` achieves up to a considerable 203% of extra cost to `Cellular`. `DEMM` is the scheme with higher energy cost due to having two interfaces enabled, since the main usage of energy is due to baseline consumption, although it saves energy compared to the exact solution of MPD2D. Nevertheless, the great gain of throughput is much higher than the extra energy cost incurred. As we observe in Figure 3.4(a), although energy consumption is significatively increased, the time required for transmitting a full piece of data is much lower, resulting in a higher energy efficiency. Figure 3.4(a) shows how much traffic the network can transport in a second (Mbps) at the cost of one energy unit (Joule). In general, `DEMM` is the most efficient scheme compared to any benchmark. `DEMM` achieves a throughput-per-energy efficiency gain from 32% compared to `Overlay` to 82% compared

to `Cellular`. Hence, the way of achieving the highest throughput and the most network efficiency is by means of the MPD2D framework.

Figure 3.4(b) sheds light on the impact of the *overlay* portion for $|\mathcal{N}| = 30$ users. We split throughput onto each connection mode in order to understand why this is the behavior. Clearly, widening the *overlay* bandwidth results into higher data rates. This shows that reuse of resources in a band with low interference helps to increase very substantially the total throughput. Moreover, with narrow *overlay* portions MPD2D and `DEMM` try to allocate D2D links over the *underlay* mode when interference is not high. However, once the *underlay* spectrum is lower than 50%, the *overlay* mode is preferred because it has higher bandwidth and much less interference problems, so that the *underlay* mode is barely used. WiFi throughput is not affected since this technology does not use licensed bands. As evident from the figure, the *overlay* portion should be maximum to achieve the highest data rates. However, widening the *overlay* bandwidth implies reducing the band for cellular connections in uplink. MPD2D reflects this fact in the decrease of cellular throughput. Here, the cellular throughput accounts for downlink and uplink, so when the *overlay* portion is 90%, the cellular mode usage is mainly due to downlink connections. Since the QoS for uplink connections cannot be much reduced, a reasonable portion for the *overlay* mode is between 30% and 40%. Otherwise, uplink traffic could easily experience low rates, for instance, in peak cases of high demand.

In Figure 3.4(c) we show the impact of changing the LTE channel bandwidth on the network performance when we apply all different D2D schemes (with $|\mathcal{N}| = 120$ users). On the one hand, we observe that `Cellular`, `Underlay` and `Overlay` schemes get to scale the utility accordingly with the bandwidth factor. This is because, as mentioned earlier, throughput takes the major part of the utility function. Since the LTE channel is the only channel used in these schemes, Figure 3.4(c) reflects the proportional scaling with the bandwidth. On the other hand, `DEMM` and `WiFi` schemes reflect that having more LTE bandwidth slightly affects the WiFi channel performance. Since `DEMM` and `WiFi` are the only schemes using the orthogonal WiFi channel, they show that widening the LTE channel bandwidth helps on cellular mode and the *Inband* D2D modes. However, utility increases less in these cases because, specially with the `WiFi` scheme, the major part of traffic goes over WiFi D2D links, which are not affected by the LTE bandwidth widening.

We also mention that we have studied also the impact of using different LTE bands, namely the 1800 MHz band (band 3) and 2600 MHz band (band 7). However, while taking higher frequencies increases the signal attenuation, it also diminishes interference issues. In practice, all the analyzed schemes provide almost the same performance (within 1% difference) on each of the LTE bands. Hence, we conclude that MPD2D framework works regardless the cellular band used.

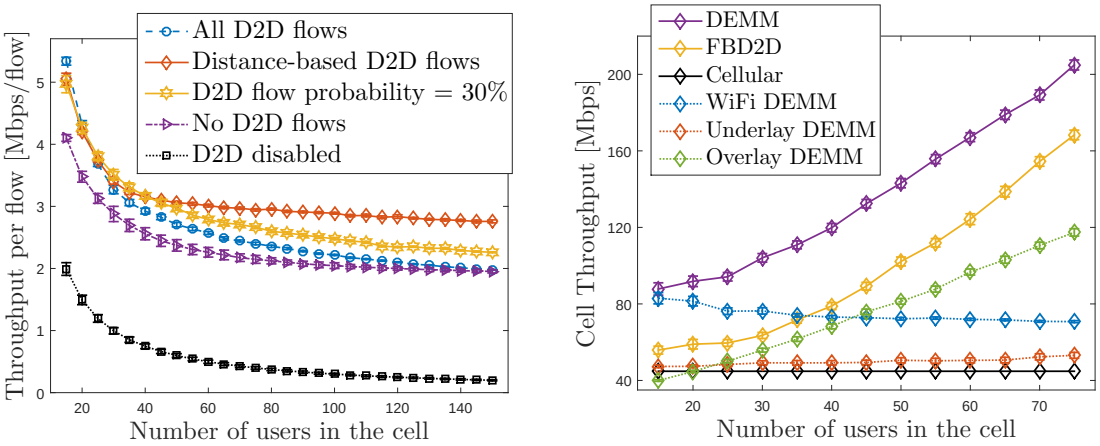
In Figure 3.5(a), we analyze the throughput per flow performance in five different setups of the D2D network. Here, we consider five scenarios in which the probability of

the establishment of D2D flows (i.e., flows that begin *and* end in a user) follows different policies. These five scenarios are, as labelled in Figure 3.5(a),

- `All D2D flows`: Every pair of users that is within the D2D range, R_{D2D}, sets a D2D flow with probability 1.
- `Distance-based D2D flows`: Every pair of users in D2D range sets a D2D flow according to a probability that decreases linearly with the distance, as studied in the rest of the chapter.
- `D2D flow probability = 30%`: Every pair of users in D2D range sets a D2D flow with a probability of 0.3.
- `No D2D flows`: No D2D flow is set, although flows may use D2D relay.
- `D2D disabled`: No D2D flows and no D2D relay.

The first four scenarios correspond to D2D-enabled networks, while the last scenario corresponds to a legacy cellular system without D2D relay. In Figure 3.5(a), first of all we observe that per-flow throughputs decay relatively fast as the number of users that share the same resources increases. However, we observe a large gap between D2D-enabled and legacy scenarios. This happens because the D2D modes considered allow to reuse transmission resources and to add extra resources (e.g., WiFi). Indeed, using D2D brings a large gain even in absence of D2D flows. Moreover, D2D flows can further exploit spectral reuse and outband communications to establish fast links without harming cellular flows and relay operations, as it is clear from the figure. This means that D2D channels offer extremely valuable and flexible resources in all cases. Figure 3.5(a) shows that the per-flow throughput achieved in the D2D-enabled scenarios is at least double and up to ten-fifteen times higher than the one of the legacy scenario. The case of `Distance-based D2D flows` behaves better than the other D2D cases reported here. Therefore, the result tells that extreme cases with all or no D2D flows, or a case in which D2D flows are blindly established without accounting for the distance, are far from being optimal in terms of throughput. Furthermore, we can observe that in non-extreme scenarios, as `Distance-based D2D flows` and `D2D flow probability = 30%`, MPD2D is able to wisely manage the resources and D2D opportunities due to lower interference levels than the `All D2D flows` scenario, and higher traffic demand than the `No D2D flows` scenario.

The analysis conducted focuses on single-cell scenarios, as formulated in the optimization (3.15). Nevertheless, the MPD2D framework can be extended to apply to multiple cells coordinated under the same network using a network controller. Such an extension comes at the cost of extra complexity, due to user population increase, the presence of cross-border D2D links, and the augmented number of links to consider for inter-cell interference management in general. Yet the complexity would be manageable



(a) Throughput per flow with different D2D flow probability distributions.

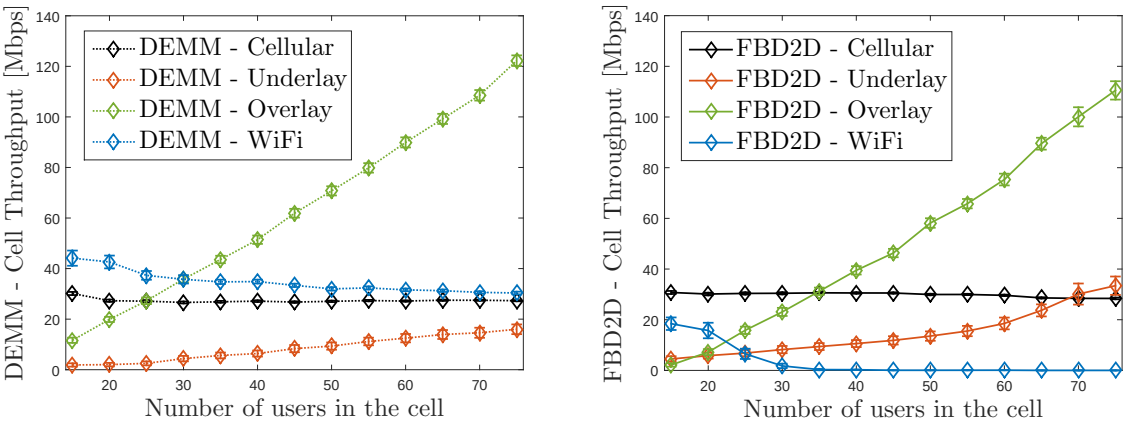
(b) Comparison of FBD2D and `DEMM`: aggregated cell throughput.

Figure 3.5: Comparison of D2D flow distributions and comparison of `DEMM` Vs FBD2D.

for few cells. However, `DIMM` will suffer severe scalability problems, whereas most of the potential interfering links added because of considering multiple cells, e.g., the ones due to users located in different and distant cells, would be efficiently labeled as non-relevant by `DEMM`. Using `DEMM` in a multi-cellular scenario would therefore result in applying `DEMM` to each cell, in which a few extra links from interfering neighboring cells are accounted for. Deploying a distributed `DEMM` would therefore be possible and efficient. In general, light complexity approaches as `DEMM` would offer viable solutions for large scale deployments.

Comparison with Floating Band D2D. Now we compare MPD2D with FBD2D [24]. FBD2D restricts each user to activate only one link for transmission and only one link for reception of data. Then, imposing that all flows from eNB were served would make very likely that most of the users activated a cellular link and missed lots of D2D chances. Therefore, we deploy a different scenario in which users set half of all the possible cellular flows in uplink and also in downlink. This deployment makes it possible to compare MPD2D and FBD2D when the latter can be used.

In Figure 3.5(b) we study the impact of network density jointly with mentioned benchmarks. Both `DEMM` and FBD2D increase throughput because they raise D2D opportunities. Clearly, `DEMM` largely outperforms FBD2D, due to the main difference between both schemes: the permission to use two interfaces for D2D. Indeed, since `WiFi` also allows two interfaces at once, FBD2D performs worse than `WiFi` until we get to a dense network and WiFi performance decays, but FBD2D still outperforms the cellular and *Inband* D2D schemes. This proves the importance in MPD2D for allowing both active interfaces (LTE and WiFi) so as to drive flows over multiple paths. Schemes like FBD2D and schemes with only one D2D mode enabled hardly split flows over multiple paths due to limited use of interfaces. For instance, `Overlay` scheme does not allow a user to send



(a) DEMM throughput splitting.

(b) FBD2D throughput splitting.

Figure 3.6: Impact of users density on DEMM and FBD2D throughput splitting, and impact of time on fairness of a dynamic cell.

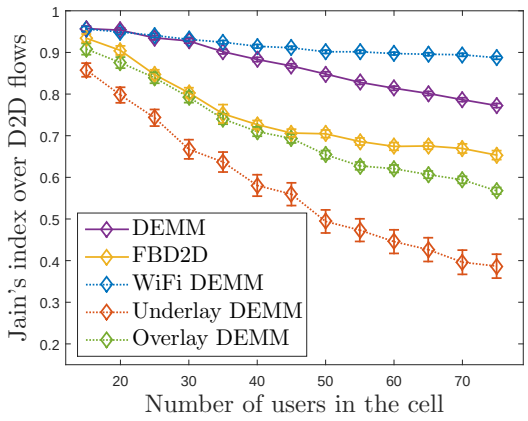
data to the eNB and also transmit through *Overlay* D2D mode because both links use the same interface (LTE). Neither FBD2D allows. In fact, FBD2D explicitly states that only one interface can be used to transfer data, as reviewed in Chapter 2. Conversely, MPD2D exploits the opportunity of enabling LTE and WiFi interfaces at once. Hence, MPD2D speeds up communications by means of splitting flows over multiple paths.

Figures 3.6(a) and 3.6(b) compare the throughput allocation from FBD2D and DEMM into all the connection modes. They reflect the advantages and drawbacks of each mode shown in Table 2.1 and why the system model of MPD2D is more flexible to achieve higher data rates. Figure 3.6(a) shows that *Inband* usage rises up until SINR limits *underlay* links. Instead, interference affects much less the *overlay* mode, and WiFi mode contributes significantly to overall throughput. In addition, Figure 3.6(b) shows that in FBD2D WiFi degrades very quickly in benefit of *Inband* modes, since FBD2D can choose only one technology. In particular, *overlay* links are more convenient due to easier SINR management. We conclude that our scheme does not need to discard any D2D mode since MPD2D can couple and use them at the same time, which results in a much higher achievable throughput.

Fairness and Satisfaction. In order to study the network satisfaction, we compute the Jain's index [94] over satisfaction values. Let $\mathcal{M} \subseteq \mathcal{N}^*$ be a subset of the nodes, the satisfaction rate over this set is:

$$J_{\mathcal{M}} = \frac{\left(\sum_{m\in\mathcal{M}} F_m(j)\right)^2}{|\mathcal{M}| \cdot \sum_{m\in\mathcal{M}} F_m(j)^2}, \tag{3.16}$$

where $F_m(j)$ are the AIIS values from Section 3.2.



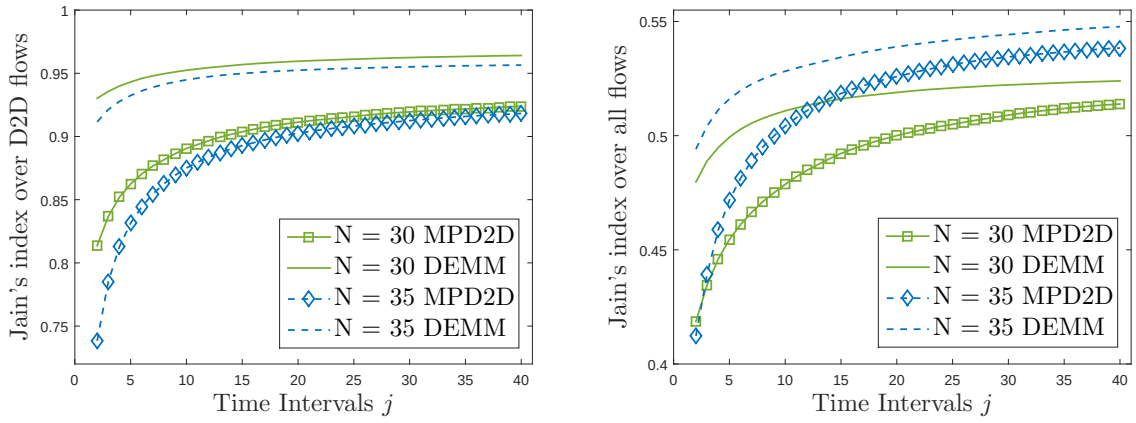
(a) Comparison of FBD2D and DEMM: Jain's index for the satisfaction metric, computed on D2D flows.

Figure 3.7: Comparison of DEMM Vs FBD2D network satisfaction.

First of all, we show in Figure 3.7(a) a comparison of network users' satisfaction fairness obtained with DEMM methods from MPD2D and with FBD2D. Here, DEMM performs quite better than FBD2D, since with DEMM the satisfaction remains between 0.77 and 0.96, while with FBD2D it decays from 0.96 to 0.65 as the network density increases. This fairness improvement comes mainly from the availability in MPD2D of combining all D2D modes in each data flow through multiple paths, as well as letting users combine more than one interface in both uplink and downlink sessions. Still, we observe that network satisfaction fairness can be improved to higher values, which motivates the integration of our *satisfaction metric*. Such an integration increases satisfaction over time to all users to much higher indices, as discussed next.

Figures 3.8(a) and 3.8(b) depict the effects of integrating the satisfaction metric developed in Section 3.2 onto the decisions over time in the cell. In Figure 3.8(a) we plot the Jain's index for D2D flows satisfaction in a static scenario, in which the set of D2D users does not change over time. Conversely, in Figure 3.8(b) we plot the Jain's index over all flows in a mobile scenario. Note that mobile users can be D2D or cellular users in different time intervals. For simplicity, we assume a fixed walking speed of 4 km/h for users moving during $T = 2$ s in random directions, for 40 consecutive time intervals.

Figures 3.8(a) and 3.8(b) show a great behavior of satisfaction over time with MPD2D and DEMM. Static D2D users increase by 18% their indices when $|\mathcal{N}| = 35$, from 0.74 up to 0.92, while the dynamic users altogether raise the satisfaction indices from 0.41 up to 0.54. In this case the Jain's index cannot be as high as in the static case, since connection opportunities are different for D2D and cellular users over time. The lesson learnt from this experiment is twofold: (*i*) multi-path is key to boost fair satisfaction of static D2D nodes to the limit, whilst (*ii*) mobile nodes experience a dramatic increase of satisfaction.



(a) Jain's index of static D2D users.

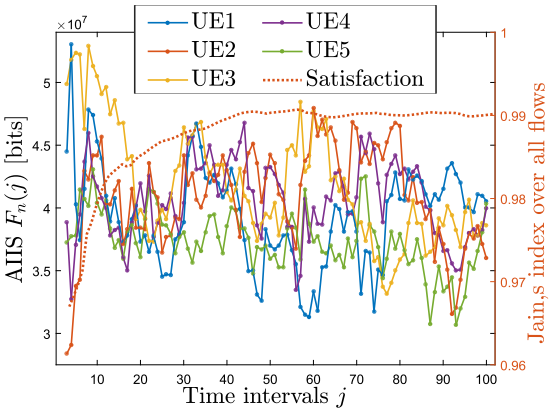
(b) Jain's index of dynamic users.

Figure 3.8: Impact of time on network satisfaction fairness.

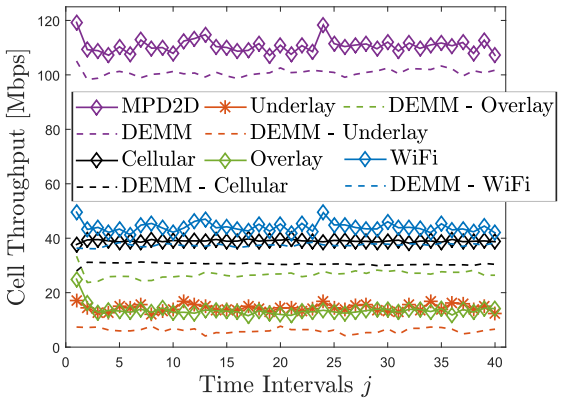
Interestingly, `DEMM` is fairer than the optimal solution of MPD2D, which is due to the fact that, as shown in Figure 3.9(b), `DEMM` uses more WiFi and less underlay links, the latter being the well-known cause of unfair behaviors among flows [95].

In Figure 3.9(a) we focus on a simpler example in order to make more visual the effects that the satisfaction metric has on average on the network, as seen in Figures 3.8(a) and 3.8(b). Here, we consider a cell with five users forming a circle inside the D2D range, so that we can establish D2D flows. Since the network is small and not complex, we apply the optimum MPD2D scheme. We locate the users in a circle in order to balance D2D connection opportunities across users. Such locations are at a distance of $R_C - 2R_{D2D}$ meters from the eNB. We take such location in order to decrease the quality of signal from cellular connections. Otherwise, when users are too close to the eNB, cellular links are selected with higher probability in this simple example and the effect of the satisfaction metric is less visible. In Figure 3.9(a) we observe the individual behavior of the satisfaction of each of the five users, as well as the performance of the Jain's index for the satisfaction of the network. The initial stage used in the experiment is a configuration under which the network utility is optimized for flows started exactly one slot before. Afterwards, the satisfaction metric gathers the experience of the flows and, following the principles of proportional fair allocation of resources, it enforces the network to increase the satisfaction while the global performance does not decay, as depicted in Figure 3.9(b). We see that, despite some users have better channel conditions and better paths for the flow demands, the satisfaction metric leads the system to make fairer decisions so that the individual satisfaction of the well-satisfied flows decays to leave space and increase satisfaction for other flows. As a result, the global system satisfaction increases over time, achieving nearly complete fairness.

Finally, in Figure 3.9(b) we analyze the cell throughput enjoyed over time in a dynamic



(a) Individual satisfaction evolution, $|\mathcal{N}| = 5$ users.



(b) Throughput in a dynamic cell, $|\mathcal{N}| = 30$.

Figure 3.9: Fairness over time, network satisfaction and throughput performance over time.

network. We show also how throughput is split into connection modes to see how the satisfaction metric affects to allocation over time. This graph is very important to see that, despite the objective function does not account for actual throughput and energy consumption, throughput does not decrease to low values as time passes by. Instead, we observe very positive results in which throughput remains stable in a reasonable variation of 15 Mbps during 80 seconds. In order to increase satisfaction, the number of *overlay* D2D links is quickly decreased in benefit of cellular connections. Still, each connection mode remains quite stable over time as the aggregated throughput does.

3.6. Lessons Learnt and Discussion

The performance assessment of the MPD2D framework proposed in this chapter unveils high potentials for D2D-based relay networks. While the availability of concurrent D2D modes optimizes the physical access to resources, MPD2D also integrates a satisfaction metric that improves users experience over time.

We have seen that it is important to properly exploit technological properties. Several user interfaces enabled at once boost the transmission efficiency, as more than one band can be used by the same user, if convenient. Also, MPD2D opportunistically allocates link connections to users in order to reuse cellular resources under the guarantee of a QoS level. In addition, the possibility of allocating traffic flows over multiple paths leads MPD2D to considerably outperform the state-of-the-art solutions.

We have seen that optimizing the network utility is not enough. As the optimization finds best link allocations according to the global interest, some users might fall under resource starvation. To avoid this issue, the DEMA scheme is key to derive a satisfaction

metric that solves unfair treatment over time. This metric is lightweight and quickly reacts to spontaneous network changes. As seen in the results, unfair network behaviours are quickly corrected thanks to this metric. In addition, the fairness correction comes at a negligible expense of network throughput.

Further improvements can be applied to MPD2D. The work described in this chapter unveils high potentials under two-hop paths. Hence, widening the search space of the optimization to multiple-hop paths might lead to better performance. However, over-saturated bands with many route hops may show small improvements. Moreover, it adds high computational complexity for the eNB to manage those hops where the eNB is not involved. Still, this is an interesting direction to look at, which we leave as future work.

4 The Millimeter-Wave Backhaul Scheduling Problem

In this chapter we explore novel RATs such as mmWave to derive an efficient relay scheme for wireless backhaul systems in dense networks. Leveraging mmWave, multiple directional communications and very high spatial reuse can be optimized in order to provide boosted traffic delivery to nodes in the edge of cellular networks.

mmWave can be used to build a wireless backhaul among a Macro Base Station (MBS) and a large number of Micro Base Stations (μBSs) spaced a few tens of meters apart, thus extending the coverage of the MBS. At the same time, the highly directional nature of mmWave makes cooperative relaying among μBSs much more useful than at lower frequencies, where interference offsets much of the potential gains. The rationale behind this approach is that the MBS is typically the bottleneck in the network, and it is therefore beneficial to offload traffic from the MBS to a nearby μBS as fast as possible. This μBS can then relay the traffic to the destination μBSs, while at the same time the MBS can already forward more traffic to the next suitable μBS. Such a relay schedule improves spatial reuse and reduces the overall time taken to distribute the traffic to the destination μBSs. The MBS can even have multiple Radio Frequency (RF) chains—the electronic device used to transmit/receive radio signals—and so communicate with more than one μBS in parallel. Figure 4.1 illustrates this approach, which is the reference scenario used in this chapter. Since it can be adapted at millisecond time scales and includes costs and advantages of beamsteering in the loop, the mmWave relay case is very different from other relay optimization problems studied in the literature and involving, e.g., WLANs, cellular and satellite networks [96] or free-space optical links [97].

The possibility of relaying and the availability of multiple antenna elements make possible communication speed-ups as described above, but also increase the complexity of scheduling data for delivery, as one needs to choose whether to relay or not, to which μBSs, and which MBS links must be used. Therefore, understanding whether scheduling data delivery is NP-hard in this context, and if so, which approximations can be guaranteed even in the limit, is fundamental to gain insight on practical challenges such as scalability. Also, in such scenario, finding out which heuristics perform well, and for which system

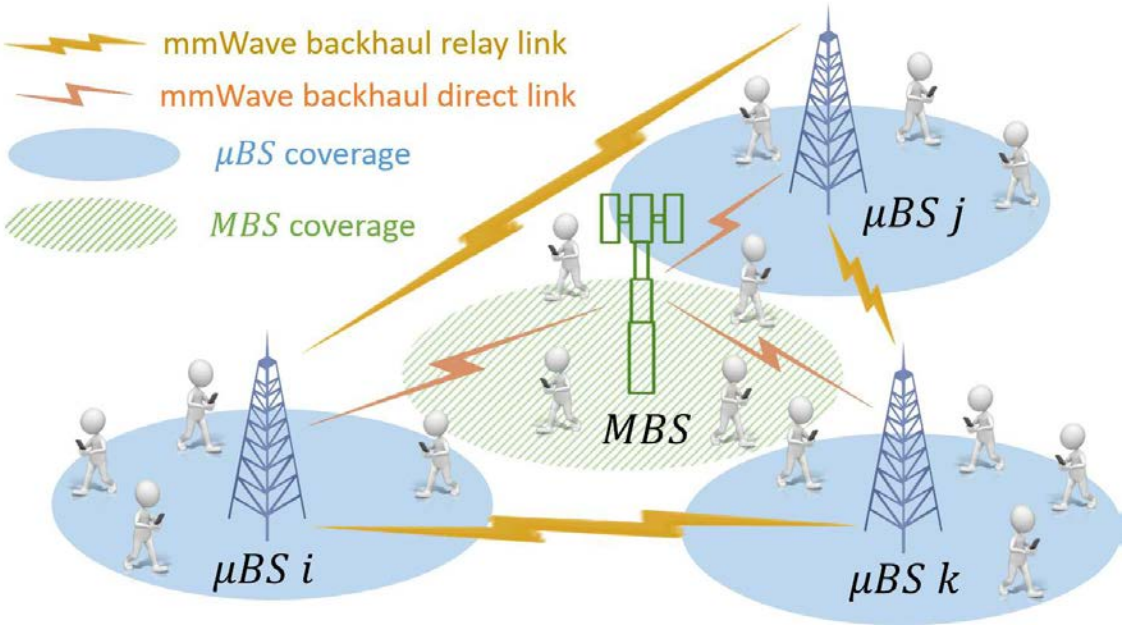


Figure 4.1: Reference scenario: mmWave backhaul network.

sizes, is crucial for practical purposes. In this chapter, we carry out such studies as follows.

Given a collection of data to deliver to a set of μBSs, we study the mmWave relay optimization problem of minimizing the time to complete the delivery, i.e. the *makespan*[1], in a network managed by an MBS. We consider both the case of interference-free links and the case of more realistic transmissions in the presence of directional cross-link interference. We call such optimization problem mmWave Backhaul Scheduling (MMWBS). Solving the problem results in a compact concurrent relaying schedule of links, which flexibly and opportunistically reuses mmWave resources over the backhaul links. However, (re-)configuring mmWave links brings with it a beam training and steering overhead that needs to be taken into account to implement a scheduling strategy that works efficiently at packet level.

In addition, we present two practical heuristics, `Greedy` and `Resched`, that approximate the MMWBS problem and are of interest for a feasible implementation in mmWave backhaul systems. We test the performance of such heuristics in comparison to optimal settings (when optima are computationally obtainable in smaller networks, due to the NP-hardness nature of the MMWBS problem), and to theoretical bounds derived. We use both synthetic data and parameters that approximate well the expected performance of future mmWave backhaul systems implementation, as well as real data measurements derived for similar mmWave scenarios. Our experimental results show that compact concurrent relaying is a powerful tool for wireless backhauling in mmWave networks.

[1]The makespan is the time needed to complete the delivery of all files. Hence, the makespan corresponds to the elapsed time until all files arrive at their destinations.

We summarize the main contributions of this chapter:

- We show that the combination of interference with the possibility of relaying makes the problem very hard, proving in Theorem 1 that not even an approximation to the optimal makespan of MMWBS with interference can be guaranteed in the worst case.
- We also show that, even without interference, MMWBS is NP-hard in Theorem 2.
- Knowing from Theorems 1 and 2 that from a theoretical standpoint we can only aim for an approximation to the optimal MMWBS schedule in interference-free channels, we find a constant approximation schedule under such assumption in Section 4.3.3. We present Algorithm 4 to compute such schedule and we provide theoretical guarantees of the approximation in Theorem 3. The above results combined expose the challenges of MMWBS.
- Theorem 3 also upper bounds the makespan of MMWBS. We establish another upper bound in Observation 1 for the natural schedule that routes all data without relaying, using only one RF chain (cf. Section 4.3.4).
- By formulating a simplified version of MMWBS in a Linear Program and using other mathematical argumentations, we prove lower bounds on the makespan of MMWBS in Section 4.3.5. We summarize our theoretical upper and lower bounds in Table 4.1.
- Leveraging the insight gained from the analysis for worst case scenarios, we design simple yet effective heuristics for MMWBS.
- We carry out realistic numerical simulations to compare the optimization, the theoretical bounds, and the heuristic approximations. The experimental evaluation shows that, on average and for small testable systems, these heuristics find near-optimal solutions, both with and without interference.

The rest of the chapter is structured as follows. We present the system model in detail in Section 4.1. We formulate the MMWBS problem in Section 4.2 as a MILP to obtain some preliminary insights. Then, we pursue a more advanced theoretical analysis in Section 4.3, where we prove that the problem is not only NP-hard but finding an approximation is also NP-hard when interference is taken into account. Hence, we also find a constant approximation schedule when interference can be neglected and theoretically bound the MMWBS problem. Section 4.4 discusses the design of `Greedy` and `Resched`, two practical heuristics that approximate well the MMWBS optimization problem. Section 4.5 reports on performance evaluation through numerical simulation. Finally, we discuss the lessons learnt in Section 4.6.

4.1. Model

We consider a backhaul system formed by an MBS s and a set $R=[1,n]$ of n static μBSs that may act as relays for s. We denote the set of nodes in the mmWave backhaul network as $V=R\cup\{s\}$, and the set of links as $E=V\times R$.

Although the main potential feature of mmWave links is the directional communication and interference mitigation for spatial reuse, it has been experimentally observed [98, 99] that commercial beam-patterns offer geometries where transmissions may potentially interfere in some regions of the space, as depicted in Figure 4.2. Such beam-patterns may have non-negligible sidelobes with high power, which indeed spoil the received signals from the μBSs positioned in the direction of such sidelobes. We model such effect in our theoretical framework by introducing an interference between transmissions via pairs of links—which could be set up arbitrarily—as follows. The binary parameter $I_{\ell_1,\ell_2}\in\{0,1\}$, known by the MBS, tells *a priori* if links ℓ_1 and ℓ_2 can be active simultaneously. Our interference model is a particular case of conflict graphs used in previous works (e.g. [100,101]), and it is justified by the fact that an active mmWave link is very sensitive to even small interference from other nearby transmissions [17]. Therefore, the binary parameter I_{ℓ_1,ℓ_2} states that links ℓ_1 and ℓ_2 cannot be active simultaneously in case their transmission beams interfere.

Time is slotted, and the capacity of each link ℓ is given as the number of bits that can be sent in one time slot, denoted as c_ℓ. Also this quantity is known by the MBS. Each link ℓ has a cost of activation $0<\alpha<1$ modelling the portion of a slot used to activate a link (antenna steering delay, potential preamble, and header overhead). Once active, a link can be used during any number of consecutive slots without incurring further activation cost. In fixed backhaul systems—as the one discussed in this chapter—the activation cost related to beam training can be saved since link end points are static. However, changing the configuration of the phased antenna array to the known setting for a link still requires a short but non-zero time. Hence, α remains positive, which is relevant for our theoretical analysis. In addition, novel designs are being considered that have non-negligible activation cost. In [102], authors test a proof-of-concept of mmWave phase shifters with miniaturized liquid crystal. Further, in [103], liquid crystal polymers are proposed as an efficient solution for future flexible 5G mmWave devices. Whereas such new antenna designs have higher activation time, they have desirable features such as higher gains and better beam shapes, that improve performance. Hence, it is important to analyze the whole framework and the system performance with positive values for the activation cost α.

We assume that each μBS, i.e., each node in R, has one RF chain, so that they can only communicate with one other node in V, and only in one direction, in any time slot. On the MBS side, we assume that up to K links from the MBS can be active in the same

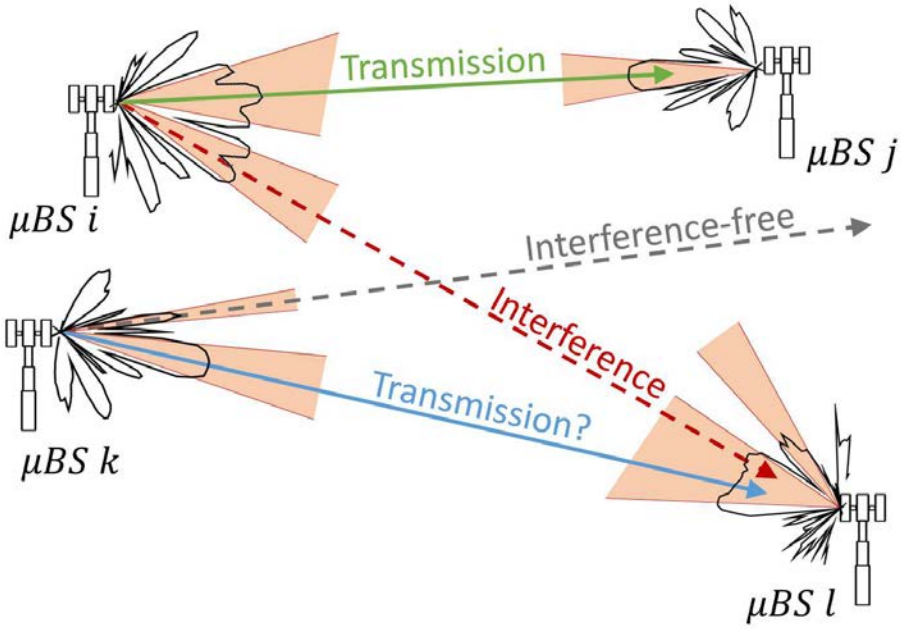


Figure 4.2: Reference scenario for the interference model.

time slot, where $K > 0$ is a parameter. That is, the MBS can communicate with several μBSs simultaneously, leveraging complicated architectures with multiple antennas and RF chains. This is to avoid a bottleneck at the MBS for file deliveries, given that the MBS is the orchestrator of all traffic arriving to the wireless backhaul network. We assume that channel state information from each pair of links is available at the MBS, following the standards defined by the 3GPP in Release 13 [104]. Hence, the network uses the sidelink transmission mode in which the network orchestrator, i.e., the MBS, manages the resources for sidelinks, as well as schedules transmissions according to the channel feedback received from μBSs control channels. Retransmissions are handled by the MAC procedure, such as the Hybrid Automatic Repeat Request (HARQ) protocol specified in the 3GPP Release 8 [105], or any other available procedure for physical resource access as CSMA/CA protocols like 802.11ad.

For each μBS $r \in R$, there is a certain amount $d_r \geq 0$ of data (in bits) whose destination is r stored at the MBS s. This data corresponds to the downlink traffic for the mobile terminals associated with that μBS, and the objective is to route it to r as quickly as possible. To this end, the MBS can send the data d_r over the direct link $\ell = (s, r)$ or via an indirect path. Path lengths are limited to two hops, i.e., there can be at most one intermediate relay r', resulting in a path $\{(s, r'), (r', r)\}$, $r' \in R \backslash \{r\}$. We leave the study of multi-hop relay to future research, as here we focus on unveiling the potential of relaying in its simplest form. We consider that R is split into two disjoint sets, $R = R^R \dot{\cup} R^D$, where

R^R is the set of μBSs that can relay (and may have their own data to receive) and R^D is the set of μBSs that are only destinations.

We define MMWBS first informally: Given a source MBS, a set of μBSs, the links between them, the interference setting, and a collection of data to deliver from source to destinations within the model described above, find a schedule of communication so that all the data is routed from the source MBS to the destination μBSs in the smallest number of time slots. The length of such a schedule is called the *makespan*. We use $\mathcal{S}$ to indicate a schedule and $|\mathcal{S}|$ for its makespan.

A formal definition of the problem, its input (communication network, the interference parameter, and collection of data) and output (a schedule of links usage) is given in the following section as an MILP.

4.2. MILP for MMWBS

We formulate the decision problem of MMWBS as an MILP. Given an integer T, we want to decide whether or not there is a communication schedule of length T slots such that all the data is routed from the source to the destinations. Accordingly, our MILP has only a set of constraints, i.e., we do not require a utility function. Hence, we search for the minimum T such that the MILP has a feasible solution. We provide now a description of the MILP. Recall that in our model, c_ℓ is the capacity of link ℓ in bits per time slot, d_r is the amount of bits with destination μBS r, and α is the link activation cost. We use the following decision variables:

- $d_{ir}(t)$ is the number of bits for destination $r \in R$ stored in node $i \in V$ at the beginning of time slot t, that is, $d_{ir}(t)$ is an integer number such that $d_{ir}(t) \geq 0$.

- $f_{\ell r}(t)$ is the fraction of the capacity of link ℓ used to send bits for destination $r \in R$ during time slot t, that is, $f_{\ell r}(t)$ is a real number such that $0 \leq f_{\ell r}(t) \leq 1$.

We now define the constraints on such variables imposed by the parameters. We start with data-flow constraints. All data is initially at the source, (Eqs. (4.1)–(4.2)), the data stored in each μBS after each time slot is updated considering the fractions of capacities used, (Eqs. (4.3)–(4.5)), and all the data must be delivered to the corresponding μBSs

after T slots (Eqs. (4.6)–(4.7)). Namely, $\forall r, i \in R, \forall j \in R^R \setminus \{r\}$, $\forall t \in [1, T]$, we have:

$$d_{sr}(1) = d_r; \tag{4.1}$$

$$d_{ir}(1) = 0; \tag{4.2}$$

$$d_{sr}(t+1) = d_{sr}(t) - f_{(s,r)r}(t)\, c_{(s,r)} - \sum_{\substack{u \in R^R \\ u \neq r}} f_{(s,u)r}(t)\, c_{(s,u)}; \tag{4.3}$$

$$d_{jr}(t+1) = d_{jr}(t) - f_{(j,r)r}(t)\, c_{(j,r)} + f_{(s,j)r}(t)\, c_{(s,j)}; \tag{4.4}$$

$$d_{rr}(t+1) = d_{rr}(t) + f_{(s,r)r}(t)\, c_{(s,r)} + \sum_{u \in R^R} f_{(u,r)r}(t)\, c_{(u,r)}; \tag{4.5}$$

$$d_{sr}(T+1) = 0; \tag{4.6}$$

$$d_{rr}(T+1) = d_r. \tag{4.7}$$

For convenience, we define a binary variable that indicates when a link is active in a given time slot:

- $a_\ell(t) \in \{0, 1\}$ is an indicator variable such that $a_\ell(t) = 1$ when link ℓ is active during time slot t.

For any link $\ell \in E$ and time slot $t \in [1, T]$,

$$a_\ell(t) \geq \sum_{r \in R} f_{\ell r}(t); \tag{4.8}$$

where Eq. (4.8) ensures that $a_\ell(t) = 1$ if some data flows through link ℓ during time slot t (recall that $f_{\ell r}(t)$ are fractions of the capacity of ℓ, hence the aggregated flow is at most 1, as constrained in Eq. (4.10)). Now, we constrain the link activations of each time slot t in order not to interfere among themselves, according to the binary interference parameter I_{ℓ_1,ℓ_2}. For all links $\ell_1, \ell_2 \in E$ and $\forall t \in [1, T]$:

$$(1 - I_{\ell_1,\ell_2}) \cdot (a_{\ell_1}(t) + a_{\ell_2}(t)) \leq 1. \tag{4.9}$$

Eq. (4.9) ensures that, in case two links ℓ_1, ℓ_2 cannot be used in the same time slot t due to interference, i.e., $I_{\ell_1,\ell_2} = 0$, there exists a constraint that permits to activate only one link in each time slot. In case links ℓ_1, ℓ_2 do not interfere, such constraint does not exist. Since the binary parameter I_{ℓ_1,ℓ_2} is an input of the problem, Eq. (4.9) is linear.

Finally, we constrain the decision variables so that the schedule obtained does not violate link capacities, including the overhead for link activation (Eq. (4.10)) and the maximum number of simultaneously active links (Eqs. (4.11)–(4.12)), due to the nodes'

- Input parameters:

  T: Number of time slots.
  d_r: Amount of bits with destination $r \in R$.
  c_ℓ: Capacity of link ℓ, given as the number of bits that can be sent in one time slot.
  α: Cost of activation relative to one time slot, $\alpha \in]0,1[$.
  I_{ℓ_1,ℓ_2}: Binary interference parameter that states if links ℓ_1, ℓ_2 can be active concurrently.
  K: Number of RF chains at the MBS.

- Decision variables:

  $d_{ir}(t)$: number of bits for destination $r \in R$ stored in node $i \in V$ at the beginning of time slot t.
  $f_{\ell r}(t)$: fraction of the capacity of link ℓ used to send bits for destination $r \in R$ during time slot t.
  $a_\ell(t)$: indicator variable such that $a_\ell(t)=1$ when link ℓ is active during time slot t.

Set of constraints, $\forall r, i \in R,\ \forall j \in R^R \setminus \{r\},\ \forall t \in [1,T]$:

- Data-flow constraints:

$$d_{sr}(1) = d_r;$$
$$d_{ir}(1) = 0;$$
$$d_{sr}(t+1) = d_{sr}(t) - f_{(s,r)r}(t)\, c_{(s,r)} - \sum_{\substack{u \in R^R \\ u \neq r}} f_{(s,u)r}(t)\, c_{(s,u)};$$
$$d_{jr}(t+1) = d_{jr}(t) - f_{(j,r)r}(t)\, c_{(j,r)} + f_{(s,j)r}(t)\, c_{(s,j)};$$
$$d_{rr}(t+1) = d_{rr}(t) + f_{(s,r)r}(t)\, c_{(s,r)} + \sum_{u \in R^R} f_{(u,r)r}(t)\, c_{(u,r)};$$
$$d_{sr}(T+1) = 0;$$
$$d_{rr}(T+1) = d_r.$$

- Activation constraints:

$$a_\ell(t) \geq \textstyle\sum_{r \in R} f_{\ell r}(t);$$
$$(1 - I_{\ell_1,\ell_2}) \cdot (a_{\ell_1}(t) + a_{\ell_2}(t)) \leq 1.$$

- Link capacity constraints:

$$\textstyle\sum_{r \in R} f_{\ell r}(t) \leq 1 - \alpha \cdot (1 - a_\ell(t-1));$$
$$a_{sr}(t) + \textstyle\sum_{j \in R} (a_{rj}(t) + a_{jr}(t)) \leq 1;$$
$$\textstyle\sum_{j \in R} a_{sj}(t) \leq K.$$

- Range constraints:

$$d_{ir}(t) \in \mathbb{Z}_{\geq 0}, \quad \forall r \in R,\ \forall i \in V,\ \forall t \in [0,T]$$
$$f_{\ell r} \in [0,1], \quad \forall l \in V \times R,\ \forall r \in R,\ \forall t \in [1,T];$$
$$a_\ell \in \{0,1\}, \quad \forall \ell \in V \times R.$$

Figure 4.3: MILP to solve the decision version of the mmWave problem.

RF chains. For any link $\ell \in E$, μBS $r \in R$, and time slot $t \in [1, T]$ we have:

$$\sum_{r \in R} f_{\ell r}(t) \leq 1 - \alpha \cdot (1 - a_{\ell}(t-1)) \,; \tag{4.10}$$

$$a_{sr}(t) + \sum_{j \in R} (a_{rj}(t) + a_{jr}(t)) \leq 1; \tag{4.11}$$

$$\sum_{j \in R} a_{sj}(t) \leq K. \tag{4.12}$$

In Figure 4.3 we present the formulation of the decision problem described in this section. Please note that we have formulated the decision version of the problem. Hence, the goal is to decide whether it is possible to route the data in T time slots, so that there is no utility function.

4.3. Theoretical Analysis

In this section, we present our theoretical study of MMWBS. We show first that, in face of interference, MMWBS cannot be approximated (Theorem 1). That is, no algorithm to compute a MMWBS schedule can have a guaranteed approximation to the optimal makespan in the worst case. The natural question that follows is what is the complexity of MMWBS without interference. We answer such question showing that even in such simpler scenario MMWBS is NP-hard (Theorem 2). In other words, we can only aim for approximations to the optimal schedule without interference. We present such algorithm (cf. Algorithm 4) and prove its approximation in Theorem 3. All these results combined expose the intrinsic challenges of solving optimally MMWBS. Nevertheless, lower and upper bounds on the makespan are of interest. Formulating a simplified version of MMWBS as a Linear Program and using other mathematical argumentations, we prove various lower bounds. The lower bound in Fact 1 shows the minimum time needed to deliver all data through the fastest interference-free links that can be active simultaneously. The lower bounds in Lemma 2 and Theorem 4 correspond to the minimum time taken by link activations even maximizing parallelism. The first is existential (corresponds to any given fixed schedule), whereas the second is universal. For comparison, we also establish upper bounds for a schedule without relaying or spatial reuse, and we bound the makespan of our approximation algorithm (Theorem 3). The upper bound in Observation 1 corresponds to delivering all data without relaying, using one MBS link at a time, and the upper bounds in Theorem 3 are makespan and approximation guarantees for Algorithm 4.

In Table 4.1 we summarize all these results, and provide specific details of each finding in the following subsections.

Table 4.1: Summary of makespan upper and lower bounds

Interference	*Makespan* $\lvert\mathcal{S}\rvert$	*cf.*
Yes	$\lvert\mathcal{S}\rvert \geq \left\lceil \alpha + \frac{\sum_{j\in R} d_j}{C} \right\rceil$ $C = \max_{\substack{R'\subseteq R: \lvert R'\rvert \leq K: \\ \forall a,b\in R': I_{(s,a),(s,b)}=1}} \sum_{i\in R'} c_{(s,i)}$	Fact 1
Independent	$\lvert\mathcal{S}\rvert \geq d + \max\left\{0, \left\lceil \frac{n'-\lvert D\rvert-(d(d-1)/2)K}{\lvert D\rvert} \right\rceil\right\}$ $d=\lceil \lvert D\rvert / K\rceil$, $D\subseteq R$ μBSs receiving directly from MBS.	Lemma 2
Independent	$\lvert\mathcal{S}\rvert \geq \left\lceil \sqrt{\frac{1}{4} + 2\left(\frac{n'}{K}+1\right)} - \frac{1}{2} \right\rceil$ $n' \leq n$ destination μBSs.	Theorem 4
Yes	$\lvert\mathcal{S}\rvert \leq \sum_{\substack{i\in R, \\ d_i\neq 0}} \left\lceil \alpha + \frac{d_i}{c_{(s,i)}} \right\rceil$	Obs. 1
No	$\lvert\mathcal{S}\rvert \leq \sum_{\substack{i\in R: \\ t_{si}>0}} \lceil \alpha + t_{si}\rceil + \frac{3}{2}\left(\left\lceil \frac{T}{1-\alpha}\right\rceil + \sqrt{3\left\lceil \frac{T}{1-\alpha}\right\rceil}\right)$ T and $\{t_{si}\}_{i\in R}$: as given by the Linear Program (LP) of Figure 4.4.	Theorem 3
No	$\frac{\lvert\mathcal{S}\rvert}{T_{OPT}} \leq \left(K+\frac{3}{2}\right)\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}}\right) + \frac{3}{2}\sqrt{3\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}^2}\right)}$ T_{OPT}: optimal makespan.	Theorem 3

4.3.1. MMWBS Cannot Be Approximated

We show here that the MMWBS Problem (with interference) cannot be approximated. We do so by reducing the *Maximum Clique Problem* [106] to MMWBS, as follows.

Theorem 1. *For all $\varepsilon > 0$, approximating the MMWBS problem to within $\sqrt{n^{1-\varepsilon}/2}$ is NP-hard.*

Proof: Consider an instance $G = (W, E_W)$ of the maximum clique problem with $|W| = n$ nodes. We build an instance of MMWBS from G as follows. The activation cost is $\alpha \leq 0.5$. The set of μBSs will be $R = W \cup \{d\}$, where $R^R = W$ and $R^D = \{d\}$. The MBS can send to up to $K = n$ μBSs simultaneously. The links from the MBS s to the μBSs in R^R interfere with each other as follows: For every $i, j \in W$, links $\ell_i = (s, i)$ and $\ell_j = (s, j)$ have $I_{\ell_i,\ell_j} = 1$ if and only if $(i, j) \in E_W$. The rest of links can only be used alone, i.e., for each link $\ell \in \{(s, d)\} \cup \{(i, j) : i, j \in R\}$, $I_{\ell,\ell'} = 0$, for all ℓ'. This interference setting guarantees that the only parallelism in the communication can come from the MBS sending to the nodes in R^R, and there can be as many simultaneous transmissions as the size C of the maximum clique in G.

There is a single file F to be sent from the MBS to d of size $f > n^2$. The links from the MBS s to the μBSs in R^R have capacity 1. The link (s, d) has extremely low capacity. The links from the μBSs in R^R to d have capacity $\frac{f}{1-\alpha}$.

It can be seen that the optimal makespan of this instance of MMWBS is $T^* = \alpha + f/C + C$, where C is the size of the maximum clique in G. The schedule uses $\alpha + f/C$ time slots[2] to get the file F to C nodes of R^R. Then, these μBSs send their corresponding portion of file to d in the next C time slots. Observe that d cannot receive anything while the file is being sent by s given the interference between links.

Let us assume there is an algorithm A with polynomial time complexity that can approximate MMWBS to within a factor $\rho = \sqrt{n^{1-\varepsilon}/2}$, for some $\varepsilon > 0$. Then, the algorithm finds a value $T \in [T^*/\rho, \rho T^*]$. Applied to the above instance of MMWBS, the obtained approximation satisfies $T \in [(f/C + C)/\rho, \rho(f/C + C)]$. Since the size of any clique is at most n, we have that $C < f/C$, and hence $T \in (\frac{f}{\rho C}, \frac{2\rho f}{C})$. This implies that $C \in (\frac{f}{\rho T}, \frac{2\rho f}{T})$. Then, we can use A to obtain an approximation of C to within $\frac{2\rho f}{T} / \frac{f}{\rho T} = 2\rho^2 = n^{1-\varepsilon}$, which is not possible unless $P = NP$ [106]. ■

4.3.2. NP-Hardness

In this section, we prove that the decision version[3] of the MMWBS problem is NP-hard, even if there is no interference and $K = 1$ [4]. The proof is via a reduction from the *Partition problem with Equal Cardinality (PEC)*: Given $2n$ natural numbers $a_1, a_2, \ldots, a_{2n}$ such that $\sum_{1 \le i \le 2n} a_i = 2B$, the question is whether there exists a partition into two subsets of n numbers each, such that the sum of each subset is exactly B. Let us consider that $a_{\min} = \min_{1 \le i \le 2n}\{a_i\} \ge \frac{B}{n+1}$.[5] This problem is NP-hard [107].

Theorem 2. *The decision version of the MMWBS problem is NP-hard for any value* $0 < \alpha < 1$ *of the link activation cost* α*, even if there is no interference and* $K = 1$.

Proof: Given an instance $\mathcal{I}$ of the PEC problem, we construct an instance $\mathcal{I}'$ for MMWBS as follows. There is the MBS s and a set $R = R^R = \{v_0, v_1, \ldots, v_{2n}\}$ of $2n + 1$ μBSs. The link capacities are $c_{(s,v_0)} = \frac{B}{1-\alpha}$; $\forall 1 \le i \le 2n : c_{(s,v_i)} = c_{(v_0,v_i)} = 1$; and $\forall 1 \le i, j \le 2n : c_{(v_i,v_j)} = 0$. For all $1 \le i \le 2n$, $d_i = a_i$ and $d_0 = 0$. The decision problem instance $\mathcal{I}'$ is whether it all data can be sent in $T = B + n + 1$ time slots.

If there is a solution for $\mathcal{I}$ then we can create a solution for $\mathcal{I}'$ by routing B data to one subset of n μBSs via v_0 and B data to the other subset of n μBSs from s directly. The communication from s to v_0 takes $\alpha + \frac{B}{B/(1-\alpha)} = 1$ time slots. The remaining communication can take place in parallel and requires $B + n$ time slots. (The

[2] We disregard ceiling and floors for simplicity.

[3] That is, whether there exists a schedule of a given makespan or not.

[4] In terms of the MILP of Section 4.2, these assumptions imply that Constraint (4.9) disappears, and Constraint (4.12) becomes $\sum_{j \in R} a_{sj}(t) \le 1$.

[5] Otherwise, we can simply add $B - (n + 1)a_{\min}$ to each value a_i.

communication of either v_0 or s with v_i requires $\lceil a_i + \alpha \rceil = a_i + 1$ time slots.) Therefore, the makespan of the schedule is $B+n+1$, which is exactly T.

We now consider the other direction. Assume that there is a solution to $\mathcal{I}'$. Observe that this solution must use μBS v_0 as relay to allow parallel communications, otherwise, the makespan would be $2B+2n\lceil\alpha\rceil = 2B+2n > T$, because the a_i numbers are natural numbers and link capacities are 1, thus, at least one extra slot is needed for each link activation. Communication from s to v_0 takes at least $\lceil\alpha\rceil = 1$ time slots just to activate the link. We claim that s must send to n μBSs via v_0 and to the other n directly. Otherwise, either s or v_0 sends to at least $n+1$ μBSs. Then, communicating with these μBSs would require at least $(n+1)(a_{\min}+1) = a_{\min}(n+1)+n+1 \geq B+n+1$ slots, since $a_{\min} \geq \frac{B}{n+1}$, and the makespan would be at least $B+n+2 > T$. This means that s sends to n μBSs via v_0 and to n μBSs directly. Finally, both v_0 and s send to their respective n μBSs B bits of data each. Otherwise, since both together send $2B$ data, one of them, say s, sends more than B bits. Let $i_1, i_2, \ldots, i_n$ be the μBSs served by s directly. Then, $\sum_{j=1}^{n} a_{i_j} > B$. Since a_{i_j} is a natural number, sending to μBS i_j takes $\lceil a_{i_j} + \alpha \rceil = a_{i_j} + 1$ slots. Hence, the makespan would be $1+\sum_{j=1}^{n}(a_{i_j}+1) > T$. ■

4.3.3. Constant-Approximation Schedule for MMWBS without Interference

In this section, we present an algorithm to obtain a schedule for the MMWBS problem that achieves a constant approximation of the optimal makespan when links do not interfere with each other. Recall from Theorem 2 that this especial case of the MMWBS problem is still NP-hard, and that in the presence of interference even approximating the optimal makespan is NP-hard (cf. Theorem 1).

In Algorithm 3 we show the `Direct Download` schedule. `Direct Download` computes a schedule that uses up to K RF chains without relaying but taking into consideration interference, and will be used in Algorithm 4.

The first step of our algorithm is to solve the LP in Figure 4.4, removing the restrictions that take interference into account (i.e., the line labelled (*)), which will be used later.

The objective function of this LP is simply to minimize the makespan, whereas the variables indicate the amount of data (flow) that has to be sent through each path (of at most two hops), and the amount of time each link has to be used. The flow and time-period constraints are the following.

1. The usage of each link does not exceed its capacity.
2. The amount of time a node is sending or receiving over any link is not more than the makespan.
3. The aggregated amount of time a set of mutually interfering links are sending or receiving is not more than the makespan. This constraint has been marked as (*) in

Algorithm 3 (Direct Download)

We define the schedule inductively in time, as follows.

For time slot $t = 1$, we select the fastest subset (up to size K) of links from the MBS (according to their downlink rate) that do not interfere. We do this one link at a time, from fast to slow, to avoid combinations. The selected set is activated in time slot $t = 1$ to download as much data as possible, while having into consideration the cost of activation of the links, α.

Then, for each time slot $t > 1$, we select first the links to μBSs that have received partially their files before t. Since they were not interfering among themselves in $t - 1$, they neither interfere in t.

Additionally, we select the fastest subset of links to μBSs that did not start their download, up to a maximum K links counting the ongoing downloads, and restricted to non-interfering links (again one by one to avoid combinations). For these newly selected μBSs we consider the cost of activation of the link, α.

Finally, in case there is some μBS out of the MBS-coverage, this μBS downloads its file through one relay, without spatial reuse.

The last time slot when some link is active following this procedure, call it t_{ub}, is an upper bound to the optimal schedule $\mathcal{S}$, i.e.: $t_{ub} \geq |\mathcal{S}|$.

Figure 4.4. Please note that obtaining the sets L described in constraint (*) is analogue to enumerating all maximal cliques in the graph $\mathcal{G} = (V \times R, \mathcal{F})$, where $(\ell_1, \ell_2) \in \mathcal{F}$ if and only if $I_{\ell_1,\ell_2} = 0$. The enumeration of maximal cliques in a graph is an NP-hard problem [108]. In practice, since the inclusion of any enumeration of maximal sets L in the restriction (*) provides a lower bound, we include a greedy enumeration of sets in the following way: given a link ℓ, we build L_ℓ as a maximal set $L_\ell \subseteq E$ such that $\ell \in L_\ell$. For this purpose, we sequentially select all links in E and check if they interfere with all links in L_ℓ. If they do, they are included in L_ℓ. Our numerical results show that this greedy enumeration of maximal sets $\{L_\ell\}_{\ell \in E}$ actually has an impact on the lower bound when the activation cost is analyzed (see Section 4.5.2).

After removing the interference restrictions, this LP has a polynomial number of restrictions. This LP can be solved optimally with standard interior-point methods [109]. For a given MMWBS input, the LP outputs the amount of data (flow) that minimizes the makespan. However, although the makespan of the LP is a lower bound on the optimal solution for the MMWBS problem, it does not solve it. In fact, the LP only outputs the period of time t_{ij} each link is active, but not how this time is distributed over slots and when links are activated. Moreover, these times do not take into account the cost of the link activation (the values t'_{ij} include only partially the activation cost). Finally, in our model, at most one link incident to each node may be active in any given time slot, but the solution obtained from the LP may violate this restriction. In our algorithm, we address these issues by modifying the schedule as follows.

We define a *vertical phase* in which first all the data held by the MBS is downloaded

- Decision variables:

  t_{ij}: time (in slots) link (i,j) is transmitting.
  f_{sij}: flow from s to j through relay i.
  f_{si}: flow from s to i without relaying.
  T: bound on the makespan.

Minimize T,
subject to:

- Flow constraints:

$$\begin{aligned}
&f_{si} + \textstyle\sum_{j \in R^R \setminus \{i\}} f_{sji} = d_i, && \forall i \in R;\\
&f_{si} + \textstyle\sum_{j \in R \setminus \{i\}} f_{sij} \leq c_{(s,i)} \cdot t_{si}, && \forall i \in R;\\
&f_{sij} \leq c_{(i,j)} \cdot t_{ij}, && \forall i \in R^R, \forall j \in R \setminus \{i\}.
\end{aligned}$$

- Disjoint-intervals constraints:

$$\begin{aligned}
&t'_{si} \triangleq t_{si}(1 + \alpha \cdot c_{(s,i)}/(\textstyle\sum_{j \in R} d_j)), && \forall i \in R;\\
&t'_{ij} \triangleq t_{ij}(1 + \alpha \cdot c_{(i,j)}/d_j), && \forall i \in R, \forall j \in R \setminus \{i\};\\
&\textstyle\sum_{i \in R} t'_{si} \leq K \cdot T;\\
&t'_{si} + \textstyle\sum_{j \in R \setminus \{i\}} (t'_{ij} + t'_{ji}) \leq T, && \forall i \in R;\\
&\textstyle\sum_{\ell \in L} t'_{\ell} \leq T, && \forall \text{maximal } L \subseteq E: \quad (*)\\
& && \forall \ell_1, \ell_2 \in L, I_{\ell_1,\ell_2} = 0.
\end{aligned}$$

- Range constraints:

$$\begin{aligned}
&t_{ij} \geq 0, && \forall i,j \in V, i \neq j;\\
&t_{is} = 0, && \forall i \in R;\\
&t_{ij} = 0, && \forall i \in R^D, j \in R;\\
&f_{sij} \geq 0, && \forall i \in R^R, \forall j \in R \setminus \{i\};\\
&f_{sij} = 0, && \forall i \in R^D, \forall j \in R \setminus \{i\};\\
&f_{si} \geq 0, && \forall i \in R.
\end{aligned}$$

Figure 4.4: LP to obtain how much data should be routed on each link and how much time each link must be active to minimize makespan. The LP does not give the schedule, i.e., a mapping from slots to link activations.

Algorithm 4 (`Constant Approximation Schedule`)

Input: an instance of the MMWBS problem.
Ouput: a mapping of data to links for each time slot.
- Solve the LP of Figure 4.4 on the given input to obtain the values $t_{si}, t_{ij}, f_{sij}, f_{si}$ for each $i, j \in R$.
- Use `Direct Download` schedule (cf. Algorithm 3) to transmit f_{si} and f_{sij} data from the MBS s to the μBS i over link (s, i) with one single link activation.
- Create a multigraph $\{V', E'\}$, where $V' = R$ and E' is a multiset of edges containing $\lceil t_{ij}/(1-\alpha) \rceil$ copies of the edge (i, j), for each $i, j \in R$.
- Run an edge-coloring algorithm on $\{V', E'\}$ and map each color to one successive time slot.
- For each of the following time slots, for each $i, j \in R$, if there is an edge (i, j) in $\{V', E'\}$ corresponding to the current time slot (color), schedule the next block of f_{sij} data, including the link-activation header if needed.

through each link $\ell = (s, r)$, $r \in R$ according to the solution of the LP. Once a downlink ℓ is active, all the data that has to go across ℓ is scheduled, hence the cost of activation is incurred only once. In this phase the schedule of the activation of the links and the transmission of the data is done as referred to in Algorithm 3 (`Direct Download`). The length of this phase is upper bounded by $\sum_{i \in R: t_{si} > 0} \lceil \alpha + t_{si} \rceil$, although in practice it is expected to be smaller due to the possibility of K parallel transmissions.

Now we define a *horizontal phase*, when the data is sent among μBSs only. To guarantee that we activate at most one link incident to each node, we create virtual links. Then, for each link that has to be active during an interval t (in slots, but maybe not an integer number of slots), we create $\lceil t/(1-\alpha) \rceil$ virtual links between the same pair of nodes. This yields a multigraph on the set of nodes R. We then apply an edge-coloring algorithm in this multigraph so that each edge incident to the same node gets a different color. We modify the schedule accordingly assigning each color to a different slot. Regarding activation costs, given that the virtual links corresponding to the same physical link might not be scheduled consecutively, we upper bound the makespan assuming that a link activates each time it is used.

We summarize the described procedure in Algorithm 4 and prove the constant approximation in Theorem 3.

Theorem 3. *Given a communication system with an MBS s, a set R of static μBSs and a collection of data to deliver from s to the μBSs within the model described in Section 4.1, the following holds:*

1. Algorithm 4 (`Constant Approximation Schedule`) outputs a schedule $\mathcal{S}$ such that

$$|\mathcal{S}| \leq \sum_{\substack{i \in R: \\ t_{si} > 0}} \lceil \alpha + t_{si} \rceil + \frac{3}{2} \left(\left\lceil \frac{T}{1-\alpha} \right\rceil + \sqrt{3 \left\lceil \frac{T}{1-\alpha} \right\rceil} \right),$$

where T and the $\{t_{si}\}_{i\in R}$ are as given by the LP of Figure 4.4.

2. *With respect to the optimal makespan T_{OPT}, the makespan $|\mathcal{S}|$ entails an approximation of at most*

$$\frac{|\mathcal{S}|}{T_{OPT}} \leq \left(K+\frac{3}{2}\right)\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}}\right) + \frac{3}{2}\sqrt{3\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}^2}\right)}.$$

3. *The running time of the Algorithm 4 is*

$$poly\Big(|R|,\ \log K + \sum_{\substack{i,j\in V:\\ c_{(i,j)}>0}} \log c_{(i,j)} + \sum_{\substack{r\in R:\\ d_{r>0}}} \log d_r,\ \sum_{r\in R}\left\lceil \frac{d_r}{1-\alpha}\right\rceil\Big).$$

Proof: The first term of the upper bound on makespan $|\mathcal{S}|$, that is $\sum_{i\in R:t_{si}>0}\lceil\alpha+t_{si}\rceil$, upper-bounds the time taken by the vertical phase, and it is simply the aggregation of data-delivery times and activations as if done sequentially, which is the worst case.

The second term corresponds to the horizontal phase and it is obtained as follows. As worst case scenario, we upper bound the cost of activations in this phase assuming that links are activated in each slot. That is, the makespan of the LP including the link-activation cost is at most $\lceil T/(1-\alpha)\rceil$. Additionally, we have to add the overhead cost of the coloring. It is known that the optimal number of colors (i.e., the chromatic index) is $\chi' \leq 3\Delta/2$ (cf. [110]), where Δ is the maximum degree of the graph. Moreover, it has been also shown in [111] how to find a coloring with $\chi'+\sqrt{9\chi'/2}$. We do not know the maximum degree of the multigraph, but we can bound it by the number of steps of the horizontal phase, which in turn is at most $\lceil T/(1-\alpha)\rceil$. Thus, using this coloring algorithm, Algorithm 4 finds a coloring of at most $3(\lceil T/(1-\alpha)\rceil + \sqrt{3\lceil T/(1-\alpha)\rceil})/2$ colors, and the claimed schedule length follows.

To see why the claimed approximation factor holds, notice that T, i.e., the makespan of the LP, is a lower bound on the optimal makespan T_{OPT}, and that $\sum_{i\in R:t_{si}>0}\lceil\alpha+t_{si}\rceil \leq K\lceil T/(1-\alpha)\rceil$. Then we have the following:

$$\begin{aligned}
&\frac{1}{T_{OPT}}\sum_{\substack{i\in R:\\ t_{si}>0}}\lceil\alpha+t_{si}\rceil + \frac{3}{2T_{OPT}}\left(\left\lceil\frac{T}{1-\alpha}\right\rceil + \sqrt{3\left\lceil\frac{T}{1-\alpha}\right\rceil}\right)\\
&\leq \left(K+\frac{3}{2}\right)\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}}\right) + \frac{3}{2}\sqrt{3\left(\frac{1}{1-\alpha}+\frac{1}{T_{OPT}^2}\right)}.
\end{aligned}$$

Finally we show the running time of Algorithm 4. The first step can be carried out with an LP solver. There is a wealth of interior-point methods that can be used for this purpose, for instance, Karmarkar's $\mathcal{O}(m^{3.5}B^2)$ algorithm [109], where m is the number of variables and B is the number of the bits in the input. Our LP has $2|R|^2$ variables and

$\log_2 K + \sum_{i,j \in V: c_{(i,j)} > 0} \log_2 c_{(i,j)} + \sum_{r \in R: d_r > 0} \log_2 d_r$ bits in the input. Hence, this step can be completed in $\mathcal{O}(|R|^7 + (\log K + \sum_{i,j \in V: c_{(i,j)} > 0} \log c_{(i,j)} + \sum_{r \in R: d_r > 0} \log d_r)^2)$ time.

For the vertical phase, adding the activation times takes $\mathcal{O}(|R|)$, as $|R|$ is the number of links outgoing from the MBS. The sorting of the μBSs takes $\mathcal{O}(|R| \cdot \log |R|)$, and the scheduling at most $\mathcal{O}(|R|^2)$. For the horizontal phase, we create the multigraph $\{R, E'\}$ where R is the set of μBSs and E' is the multiset of edges. For each μBS $r \in R$ the number of incoming virtual links is $\lceil d_r/(1-\alpha) \rceil$. Then, $|E'| \leq \sum_{r \in R} \lceil d_r/(1-\alpha) \rceil$ and the total time to create the multigraph is in $\mathcal{O}(|R| + \sum_{r \in R} \lceil d_r/(1-\alpha) \rceil)$. The next step is the edge-coloring algorithm of [111], which runs in $\text{poly}(\nu, \log \mu)$ time for a multigraph of ν nodes and maximum multiplicity μ. Then, this step takes $\text{poly}(|R|, \log \max_{r \in R} \lceil d_r/(1-\alpha) \rceil))$. Combining all the running times, the claim follows. ■

4.3.4. Makespan Upper Bound

An upper bound on the MMWBS makespan is given by a scheme that routes all data without relaying or spatial reuse.

Observation 1. *Given a communication system with an MBS s and a set R of n static μBSs, and a collection of data to deliver from s to the μBSs within the model described in Section 4.1, there exists a schedule $\mathcal{S}$ such that*

$$|\mathcal{S}| \leq \sum_{\substack{i \in R, \\ d_i \neq 0 \\ c_{(s,i)} \neq 0}} \left\lceil \alpha + \frac{d_i}{c_{(s,i)}} \right\rceil + \sum_{\substack{i \in R, \\ d_i \neq 0, \\ c_{(s,i)} = 0}} \left\lceil \alpha + \frac{d_i}{c_{(s,j_i)}} \right\rceil + \left\lceil \alpha + \frac{d_i}{c_{(j_i,i)}} \right\rceil. \tag{4.13}$$

The makespan in Observation 1 comes from a schedule that delivers data sequentially to μBSs under MBS-coverage using only one RF chain in each time slot (first term). For those μBS i out of MBS-coverage, it sequentially considers indirect paths by relaying over a μBS j_i that relays d_i to i through the fastest path, but without spatial reuse (second term). Therefore, interference has no impact on this bound.

4.3.5. Lower Bounds on the Makespan

To solve the MMWBS problem, the MBS has to send to μBSs all the data using a scheduling $\mathcal{S}$. Let $|\mathcal{S}|$ be the makespan of $\mathcal{S}$ in time slots. Sending all the data through the fastest links outgoing from s, using up to K interference-free links concurrently, gives the following lower bound.

Fact 1. *Given a communication system with an MBS s and a set R of μBSs, and a collection of data to deliver from s to the μBSs within the model from Section 4.1, in the*

presence of interference, consider a schedule $\mathcal{S}$ that solves the MMWBS problem. Let

$$C = \max_{\substack{R' \subseteq R : |R'| \leq K : \\ \forall a,b \in R' : I_{(s,a),(s,b)} = 1}} \sum_{i \in R'} c_{(s,i)}.$$

Then, the length of $\mathcal{S}$ (i.e., the makespan) is as follows:

$$|\mathcal{S}| \geq \left\lceil \alpha + \frac{\sum_{j \in R} d_j}{C} \right\rceil. \tag{4.14}$$

Another lower bound, based on link activation, is proved for a given schedule in Lemma 2, and a universal lower bound (for any schedule) is given in Theorem 4. These bounds are relevant when the amount of data to send is small and the makespan is dominated by link activations.

Lemma 2. *Given a communication system with an MBS s, a set R of n μBSs, and a collection of data to deliver from s to $n' \leq n$ μBSs within the model from Section 4.1, consider a schedule $\mathcal{S}$ that solves the MMWBS problem. If the set of μBSs that receive data directly from s is $D \subseteq R$ then, even without interference,*

$$|\mathcal{S}| \geq d + \max\left\{0, \quad \left\lceil \frac{n' - |D| - (d(d-1)/2)\, K}{|D|} \right\rceil \right\}, \text{ with } d = \lceil |D|/K \rceil.$$

Proof: Let the data to be delivered be called simply *data*. Let a μBS that has data, for itself or for other nodes, be called *informed*. Consider the sequence of time slots $t_1, t_2, \ldots, t_d$ when the μBSs in D are informed (possibly interleaved with other time slots when no μBS in D is informed). Recall that D is defined to be the set of μBSs that receive directly from the MBS. Then, in each time slot, at most K new μBSs in D may be informed, and it is $d \geq \lceil |D|/K \rceil$. For $1 \leq i \leq d$, let $D(t_i)$ be the subset of μBSs in D that have been informed by time t_i. Then, $|D(t_i)| \leq iK$ for $1 \leq i < d$, and $|D(t_d)| = |D|$.

Let $\mathcal{I}$ be the set of μBSs that do not receive directly from s in the schedule $\mathcal{S}$. For any time slot t, let $\mathcal{I}(t)$ be the subset of μBSs in $\mathcal{I}$ informed during time slot t. Then, given that μBSs in $\mathcal{I}$ are only informed by the μBSs in D, we have that

$$|\mathcal{I}(t)| \leq \begin{cases} 0 & \text{for } t \leq t_1; \\ |D(t_i)| & \text{for } t_i < t \leq t_{i+1} \text{ and } 1 \leq i < d; \\ |D| & \text{for } t > t_d. \end{cases}$$

Since $|D(t_i)| \leq |D|$ for all $1 \leq i \leq d$, to prove the lower bound we assume as a worst case that all μBSs in D are informed in the first $d = \lceil |D|/K \rceil$ time slots. Then, the sequence of numbers of μBSs informed along time slots $1, 2, \ldots, d$, is $|D(t_1)|, |D(t_2)| + |\mathcal{I}(t_2)|, \ldots, |D| +$

$\sum_{i=2}^{d} |\mathcal{I}(t_i)|$. So, at the end of slot d, the number of informed μBSs is

$$|D| + \sum_{i=2}^{d} |\mathcal{I}(t_i)| \leq |D| + \sum_{i=2}^{d} (i-1)K = |D| + \frac{d(d-1)}{2} K. \tag{4.15}$$

From slot t_d+1, at most $|D|$ new μBSs are informed in each slot. Thus, to inform remaining μBSs (if any), we need at least $\lceil \left(n' - |D| - \frac{Kd}{2}(d-1) \right) / |D| \rceil$ additional slots. Adding this time to the previous d time slots, the claim follows. ■

Theorem 4. *Given a communication system with an MBS s, a set R of n μBSs, and a collection of data to deliver from s to $n' \leq n$ μBSs within the model from Section 4.1, consider a schedule $\mathcal{S}$ that solves the MMWBS problem. Then,*

$$|\mathcal{S}| \geq \left\lceil \sqrt{1/4 + 2\,(n'/K + 1)} - 1/2 \right\rceil.$$

Proof: Consider the maximum number of μBSs that may be informed in slots $1, 2, \ldots$. In the first time slot at most K μBSs may be informed. In the second slot at most K μBSs may be informed directly and another K may be informed by relaying. We continue the same analysis to compute the first time slot t when $tK + \sum_{i=2}^{t-1} iK \geq n'$, i.e., when

$$t^2 + t - 2(n'/K + 1) \geq 0.$$

Solving the quadratic equation the claim holds. ■

Another lower bound on the makespan is given by the LP of Figure 4.4 (see Section 4.3.3 for an explanation of the constraints). The objective function here is simply to minimize the makespan, whereas the variables indicate the amount of data (flow) that has to be sent through each path (of at most two hops), and the amount of time that each link is used.

Notice that the formulation does not restrict the temporal order in which the links must be used (as opposed to the MILP of Section 4.2). For instance, data that is being relayed can be delivered to a given destination only after reaching the relay. This LP does not restrict such temporal order. Therefore, the makespan obtained from the solution of this LP is only a lower bound on the optimal makespan. The reason being that, anyway, flow and time-period restrictions have to be observed, but the optimal makespan for MMWBS could be even larger after additionally restricting the order in which links are used.

In the experimental evaluation, we compare our algorithms with the maximum of the lower bounds obtained in Fact 1, Theorem 4, and the solution of the LP in Figure 4.4.

4.4. Heuristics

4.4.1. `Greedy` Heuristic

We present first a simple greedy heuristic that is based on using those links that are faster as soon as they are available. Specifically, the schedule of transmissions with `Greedy` is built as follows.

At any time slot t in which an RF chain k in the MBS s is available (e.g., initially or when it completes sending data to a μBS), it schedules a new transmission across the fastest link (s, i) from s to any available μBS $i \in R^R$. The data $d_j > 0$ sent in this transmission will be for the available μBS j that has the fastest link (i, j), among the μBSs whose data is still at s. Such scheduling begins in the first time slot later or same than t where the download of d_j by link (s, i) and the relaying of d_j by link (i, j) does not interfere with other allocated transmissions. Nodes s and i are then unavailable for a period of $\lceil \alpha + d_j/c_{si} \rceil$ time slots in case link (s, i) was not used through RF chain k in the time slot prior to this link allocation, or for a period of $\lceil d_j/c_{si} \rceil$ otherwise. When the data d_j is completely received at i, the MBS becomes available again in the next time slot and at the same time the data is forwarded to j via the link (i, j). μBSs i and j are then unavailable for a period of length $\lceil \alpha + d_j/c_{ij} \rceil$ time slots. Here, the activation cost α is considered since link (i, j) cannot be active in any previous time slot. In case at some point an RF chain in the MBS becomes available and there is only one available μBS i (because all the others already have their data or because they are relaying/busy), the MBS schedules a direct download of data d_i for μBS i in the earliest time slot that does not interfere.

Complexity. At each iteration, while there is data to be served, `Greedy` takes the fastest available μBS and sends it data d_j for the fastest available neighbour μBS that still does not have its data. Hence, we enable parallel transmission as soon as possible, as intended in our mmWave backhaul with relaying. In case this is not possible, `Greedy` schedules a direct download. Since checking interference issues only consists into logical checks, it takes at most n iterations, one per μBS waiting for a file, to decide a final schedule. Thus, the computational complexity of this algorithm is $\mathcal{O}(n)$.

4.4.2. `Resched` Heuristic

The second heuristic is based on reschededuling an initial communication assignment in order to iteratively improve the makespan at each step until no further improvement is possible. We call this heuristic `Resched`.

We consider an initial feasible schedule provided by `Direct Download` (cf. Algorithm 3), which consists of sequential direct downloads without relaying from the MBS to the μBSs under MBS-coverage that have a file to be served, while using all K RF

chains and avoiding interfered connections, and sequential relayed downloads for those μBSs out of MBS-coverage. The set of μBSs that have to receive a file is sorted from lowest to highest delivery time. At this point, every μBS $r \in R$ such that $d_r > 0$ is scheduled through an RF chain $k_r \leq K$ and at a time slot in which it begins its download, which we call its initiation instant: t_{sr}. For those $r' \in R^R$ such that $d_{r'} = 0$ we let $t_{sr'}$ unset. Then, we iteratively modify such assignment by rescheduling the transmissions, taking advantage of relaying.

At each iteration, we take the μBS $u \in R$ such that $d_u > 0$ that receives its file the latest and that has been tried to be reallocated less times. Then, we reallocate the path transmission of its data, d_u. For such reallocation, we search for a μBS $r \in R^R$ that may potentially relay the data d_u to u. Thus, we check each μBS $r \in R^R$ in the order of the sorted list of μBSs and take the one that reduces most the current makespan and has no interference issues (this is, the new retransmission of the data d_u cannot be scheduled through links that interfere with the already allocated transmissions of the other files of the system). The way in which the retransmission of data d_u is allocated is the following:

We take the initiation instant of r, t_{sr}. Let τ_r^δ be the number of time slots the μBS r is downloading files and let τ_r^ρ be the number of time slots the μBS r is relaying files, according to the current schedule. In case r is already downloading data from the MBS, i.e. $\tau_r^\delta > 0$, we take the RF chain k_r through which such download is scheduled, and define a binary indicator $\xi_r = 1$. Otherwise, i.e. if $\tau_r^\delta = 0$, we select an RF chain $k_r \leq K$ in the MBS such that k_r stops being used in the earliest time slot t and define the binary indicator $\xi_r = 0$. Then, we take $t_{sr} = t$. The μBS r begins to download in time slot t_{sr} the data d_u through the RF chain k_r. Since data d_u is now downloaded by r, instead of by u, the RF chain k_u gets free for those times slots corresponding to the direct download, $[t_{su}, t_{su} + \lceil \alpha + d_u/c_{(s,u)} \rceil - 1]$, so that other transmissions can be allocated in such time slots later in the heuristic. The download of d_u will take place through RF chain k_r in the time slots

$$\Big[t_{sr} + \tau_r^\delta - 1, \quad t_{sr} + \tau_r^\delta + \lceil \alpha \cdot \xi_r + d_u/c_{(s,r)} \rceil - 1\Big]. \tag{4.16}$$

Then, we update τ_r^δ to $\tau_r^\delta = \tau_r^\delta + \lceil \alpha \cdot \xi_r + d_u/c_{(s,r)} \rceil$. Please note that, in case r was already downloading data before allocating to it the download of data d_u, we do not consider the activation cost α, while in the opposite case we do[6]. Once r downloads all the files currently allocated to it, r relays the data d_u to u in the time slots

$$\Big[t_{sr} + \tau_r^\delta + \tau_r^\rho - 1, \quad t_{sr} + \tau_r^\delta + \tau_r^\rho + \lceil \alpha + d_u/c_{(r,u)} \rceil - 1\Big]. \tag{4.17}$$

[6]We consider that every relay $r \in R^R$ first downloads all the data allocated to it, so that the activation cost with the MBS is considered only once. After all the downloads by r end, r begins to relay the files with the other μBSs allocated to r.

Then, we update τ_r^ρ to $\tau_r^\rho = \tau^\rho + \lceil \alpha + d_u/c_{(r,u)} \rceil$. Regardless the case, α has to be considered when updating τ_r^ρ because link (r, u) could not have been activated before.

The reallocation described for data d_u clearly affects the scheduling of those files downloaded through RF chain k_r. Since now r has to use more time slots to download one file more, the transmissions beginning later than t_{sr} through RF chain k_r must be delayed, as well as the relaying of such files. Thus, for all $r' \in R$ such that $t_{sr'} > t_{sr}$, the initiation instant of r' is updated to:

$$t_{sr'} = t_{sr'} + \lceil \alpha \cdot \xi_r + d_u/c_{(s,r)} \rceil. \tag{4.18}$$

Thus, all the time slot intervals used for relaying are delayed as well, according to the new value of $t_{sr'}$, and the current makespan is modified.

If now the makespan is shorter than before and there are not interference issues, we keep the new schedule. Otherwise, we discard such reallocation of μBS u and try to relay data d_u through a different relay $r \in R^R$ not selected for u yet. If all relays in R^R have already been selected for u without success, another iteration starts for the relaying of data from the next μBS in the sorted list of μBSs, and u is replaced the last in the sorted list.

`Resched` ends when all non-reallocated nodes waiting for a file have been tried to be rescheduled without success.

Complexity. `Resched` begins with an initial feasible allocation and then, it checks μBS by μBS with a file in the MBS if its download can be rescheduled to reduce the makespan. For each μBS, the heuristic checks up to $|R^R| \leq n$ possibilities for its rescheduling. Thus, since checking interference issues only consists into logical checks, the computational complexity is $\mathcal{O}(n^2)$.

4.5. Experiments

We perform experiments in scenarios that accurately reproduce real mmWave communication systems. We consider an MBS and a set of μBSs, as described in Section 4.1. We perform experiments with different choices for sets R^R and R^D where we analyze the behaviour of our algorithms, as well as different numbers of RF chains in the MBS. Also, we model interference in the network based on a model for mmWave antenna patterns that measures the radiating beamwidths, as detailed later. To shed light on scenarios based on real measured rates and beam-patterns, we perform simulations in which link Signal-to-Noise Ratio (SNR), link rates and interference are obtained from real experiments, as detailed later.

To collect consistent statistics, we simulate each scenario 1000 times. We show in each plot the lower bounds detailed in Section 4.3.5. Nevertheless, we compact them onto the

maximum lower bound that we get on each experiment, i.e., we select the lower bound with better guarantees for the makespan. Regarding upper bounds, here we mention that `Constant Approximation Schedule` (cf. Algorithm 4) provides theoretical guarantees for the makespan in absence of interference, as proved in Theorem 3. However, the `Direct Download` schedule described in Algorithm 3 practically provides better upper bounds in any case (either in absence or presence of interference), although it does not give theoretic guarantees. Hence, here we show the makespan provided by `Direct Download`.

The labels of figures legends refer to the makespan derived with the following schemes:

- **Upper Bound:** Makespan resulting from `Direct Download` (cf. Algorithm 3) and Observation 1.
- **Lower Bound:** Longest makespan from Fact 1, Theorem 4 and Figure 4.4.
- **Resched:** Makespan obtained with `Resched` heuristic.
- **Greedy:** Makespan obtained with `Greedy` heuristic.
- **Optimum:** Optimum makespan obtained by solving the MILP for MMWBS problem derived in Section 4.2.

4.5.1. Experimental Setup

In the experiments, we consider that the MBS has data of random length for each μBS, drawn from a truncated exponential distribution ranging from 1 MB to 80 MB, with an average of 10 MB. We deploy a circular cell of radius R_C centered in the MBS and place uniformly at random n μBSs inside the cell. We consider a fixed transmit power of $P_t = 30$ dBm, and fixed antenna gains of $G_t = 25$ dB and $G_r = 25$ dB for transmitter and receiver respectively. According to the Friis equation, the power received in dB, P_r is $P_r[dB] = P_t + G_t + G_r + 10\eta \log_{10}\left(\frac{\lambda}{4\pi\delta}\right) - 5$, where $\eta = 2$ is the path loss exponent in free space, λ is the wavelength in meters for 60 GHz carrier frequency, and δ is the distance between transmitter and receiver. Besides, we subtract an implementation loss of 5 dB. The thermal noise power of Johnson-Nyquist in dBm is: $P_N[dBm] = -174 + 10\log_{10}(W)$, where $W = 2.16 \cdot 10^9$ is the bandwidth in Hz used for transmission. We amplify this noise by the receiver noise factor of 40 dB, so the actual noise in dBm in the receiver is $N[dBm] = P_N + 40 = -174 + 10\log_{10}(W) + 40$. The achieved signal-to-noise ratio is $SNR = 10^{\frac{P_r - N + 30}{10}}$, and the electronic sensitivity S in dBm at the receiver is $S[dBm] = 10\log_{10}(k(T_a + T_{Rx})W \cdot SNR) + 30$, where k is the Boltzmann constant, T_a is the noise temperature in Kelvin of the antenna at the input of the receiver, T_{Rx} is the noise temperature in Kelvin of the receiver referred to its input, and W is again bandwidth. We use $T_a = T_{Rx} = 290$ K.

We use the link rates resulting from the above computation and corresponding to the Single-Carrier Physical (SCPHY) modulation and coding schemes of [112], which have been implemented in commercial mmWave devices. SCPHY rates range from 385 Mb/s to 4620 Mb/s. We further use an Equivalent Isotropically Radiated Power (EIRP) of 55 dBm. This makes possible to find theoretically feasible links with $R_C = 35$ m, as we adopt.

The time slot is fixed to $T_s = 10$ ms, and the activation time, namely A_t, of every link is 1.0349 ms, unless otherwise specified. This activation time corresponds to the antenna steering time spent to explore a finite number of sectors, as specified, e.g., in the IEEE 802.11ad amendment [112]. Hence, this time is composed of a preamble and a header of 4291 and 4654 nanoseconds each, plus the time needed to steer the antenna towards the intended receiver and the intended transmitter. Here we assume that base stations have arrays of antennas of $N_A = 32$ sectors. Since the time to transmit the Sector Sweep Frame (SSW) is 15.76 μs, and both transmitter and receiver need to steer their beams one after the other, the final activation time will be:

$$A_t = 2N_A \cdot 15.76\ \mu\text{s} + 18.2\ \mu\text{s} + 4.29\ \mu\text{s} + 4.65\ \mu\text{s} \approx 1.0349\ \text{ms}. \tag{4.19}$$

The first term in Eq. (4.19) corresponds to the N_A SSWs from the transmitter and N_A SSWs from the receiver. The second term corresponds to one feedback frame from the transmitter, and the remaining terms are the preamble and header duration, respectively. Since we assume slot length normalized to 1, we have that every link has an activation cost of $\alpha = A_t/T_s = 0.10349$.

We discuss two cases of the model described in Section 4.1:

- Full Network: We consider that the full network helps to relay data, i.e., $R^R = R$ and $R^D = \emptyset$, and that all the μBS $r \in R$ have data of size $d_r > 0$ to download. This case fits the main purpose of this research, when a mmWave backhaul network is deployed to obtain the files for the user equipments served by the μBSs as quickly as possible. Thus, relaying is enabled in the full network.

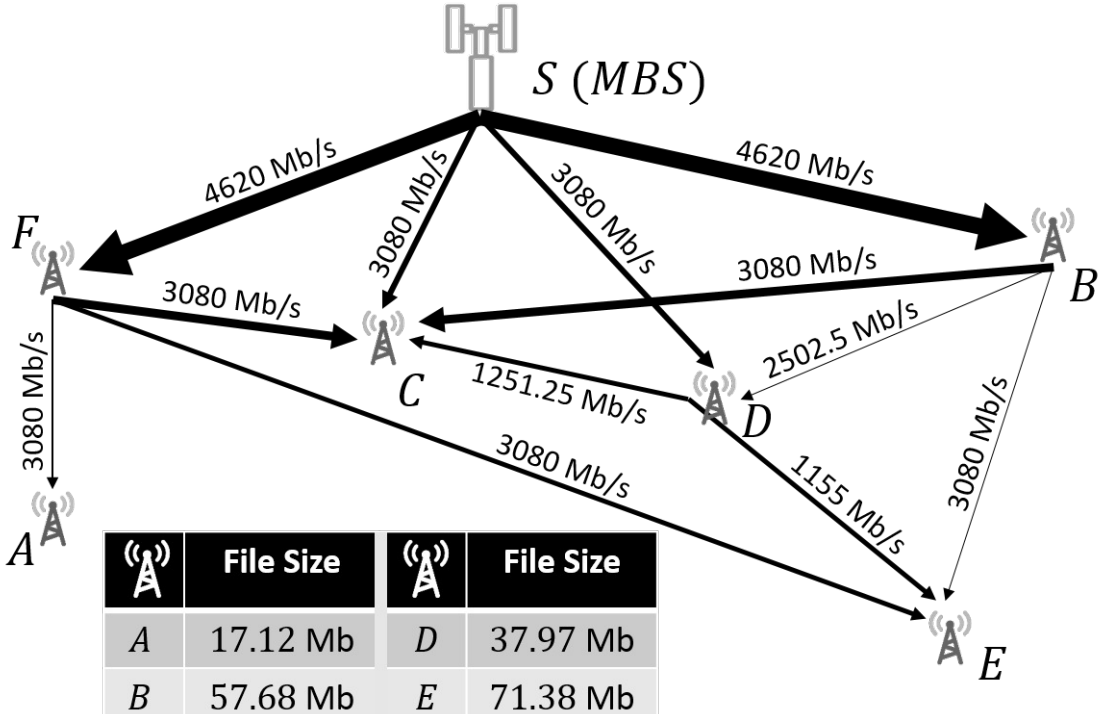
- Small Cell Network: We consider that a set of μBSs act as relays whose only purpose is to help the network by relaying data. They do not have data for themselves, so $\emptyset \neq R^R \subsetneq R$ and $d_r = 0, \forall r \in R^R$. Although this research is intended for a mmWave wireless backhaul network, this case more generically represents and sheds light on scenarios where a mmWave access network is deployed, so that there may be mmWave devices willing to help the network through relaying but do not claim any file.

Furthermore, we also study scenarios under the effects of interference, as described in the model of Section 4.1. In order to capture such effects, we consider an interference model based on antenna patterns with one main lobe of a given Half-Power Beamwidth

(HPBW), and sidelobes with a power lower than such HPBW. This model represents real directional antennas that may radiate and interfere only within an angle given by the HPBW. Such assumption is based on a simple analytical model [113] which characterizes the electric field of an antenna as a function of the beamwidth. As depicted in Figure 4.2, actual beam-patterns may radiate non-negligible power in sidelobe directions. The beam-patterns depicted in the figure correspond to actual patterns experimentally obtained in a recent work [98, 99]. In our experiments, we use HPBW values large enough to account for sidelobes. In any case, the interference matrix I (i.e., parameter I_{ℓ_1,ℓ_2}) is assumed as an input of the problem, so that the performance of our algorithms is not affected by the exact shape of the antenna patterns. Moreover, since the deployment of μBSs is uniformly random, the resulting average results are the same as if an arbitrary number of sidelobes were considered. In [98, 99], authors have observed that a typical aggregate radiating width for commercial antennas is around 22.5 degrees, which approximately corresponds to $\pi/8$ radians, as we mainly adopt in our numerical evaluation. In addition, we show at the end of the section results in which we use measured data for the exact shape of beam-patterns and link rates obtained from a commercial mmWave device: the TP-Link Talon AD7200 router. While this mmWave device is for indoor use and mmWave backhaul has somewhat different characteristics (more refined beam-patterns, higher rates, ...), the underlying RF technology and specifically the phased arrays are similar enough for these measurements to give meaningful results. The exact shape of these beam-patterns is available at [114] and the data can be downloaded at [115]. In order to build the real binary interference map we need to know which beam-pattern each μBS will use with each of its neighbors. Hence, we simulate the beam-training of links based on the link SNR: when a link is trained, each of the μBSs involved tests all of its beam-patterns and chooses the one that provides the highest SNR. The achievable link rates of these mmWave devices have been investigated in [116], from where we obtain the link data rate based on the distance between two devices. This allows to study our framework for realistic relaying and spatial reuse scenarios. We further provide a performance evaluation based on synthetic and modeled beam-patterns that represent future backhaul applications using better hardware.

4.5.2. Numerical Results

Here we present multiple results of numerical experiments. We use `R2018a` version of MATLAB in order to simulate channel conditions, packet sizes, interference topology and positions of μBSs. We use the CPLEX optimizer to find optimum values for small instances of the MMWBS problem. In fact, it is hard to obtain optimal solutions from the MILP formulation presented in Section 4.1, due to the NP-hardness proven in Section 4.3.2. Thus, optimum values are only available for small numbers of μBSs, which have been obtained through exhaustive search methods as Branch & Bound [117]. In the



	File Size		File Size
A	17.12 Mb	D	37.97 Mb
B	57.68 Mb	E	71.38 Mb
C	239.16 Mb	F	26.24 Mb

Scheduling:

$t=1$	$t=2$	$t=3$	$t=4$	$t=5$	$t=6$	$t=7$	$t=8$	$t=9$	$t=10$	$t=11$	$t=12$
$S \to F$	$S \to$ B	$S \to F$	$S \to B$	$S \to D$	$S \to D$	$S \to B$	$S \to F$	$S \to F$	$S \to C$	$S \to C$	$S \to B$
A: 17.12	C: 38.04	C: 41.42	C: 41.42	C: 15.08	D: 20.45	B: 16.23	C: 25.65	C: 1.96	C: 27.61	C: 30.8	B: 41.42
C: 13.80	$F \to A$	$B \to C$	$F \to C$	D: 12.54	E: 10.35	C: 3.37	F: 15.76	E: 44.24	$B \to D$	$F \to E$	$D \to E$
F: 10.48	A: 17.12	C: 27.61	C: 27.61	$B \to C$	$F \to C$	D: 4.99	$B \to C$	$D \to C$	D: 4.99	E: 16.63	E: 10.35
				C: 27.61	C: 27.61	E: 16.78	C: 27.61	C: 3.86	$F \to E$		$F \to C$
Upper Bound: 17 *TS*						$D \to C$		$B \to E$	E: 27.61		C: 27.61
Lower Bound: 10 *TS*						C: 11.22		E: 16.78			

Figure 4.5: Example of optimal scheduling with 6 μBSs and one MBS with single RF chain ($K = 1$). The figure shows the logical topology, the set of links used and their utilization, the scheduling and the makespan with its bounds.

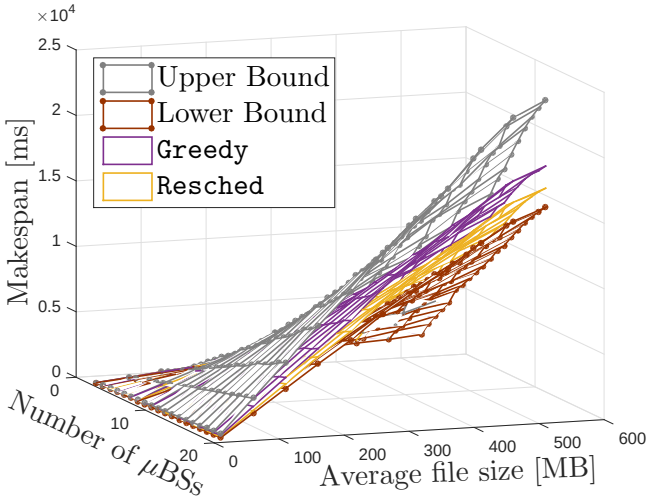


Figure 4.6: Makespan in the full network case, with $K = 1$ RF chain and without interference: Impact of the size of the network, n, and the average file size.

figures shown in this section, error bars represent 95% confidence intervals centered on the average extracted from 1000 simulations.

We show in Figure 4.5 a simple example of an optimal schedule. Here, we consider $n = 6$ μBSs, labeled from A to F, and an MBS with only one RF chain ($K = 1$). The link rates shown are extracted from the SCPHY modulation and coding schemes defined in the standard. The rate corresponds to the SNR experienced by the receiver according to the power and noise computed as described in Section 4.5.1. Therefore, rates depend on the distance between transmitter and receiver, and on the adopted channel model. The upper part of the figure shows the logical topology (not the physical one), the links that are used, the data rate of links (label next to the link), the utilization of each link (thickness of the lines), and the size of downloaded data (in the table on the left). The bottom part of the figure shows the optimal schedule, which corresponds to a makespan of 12 time slots (TS in the figure) for this example, and is very close to the lower bound of 10 slots indicated above the scheduling table. This makespan is a 30% lower than the upper bound of 17 slots of the schedule with no relaying. Hence, end-user demands can be served much faster at the serving μBS. We also show at each time slot which links have been scheduled (marked with an arrow), which μBSs are intended to receive the data sent over such links (marked with the μBS label), and how much data in Mb is sent for the intended destination. In what follows, we show that this makespan improvement is the general behavior, thus proving the importance of studying relay and spatial reuse featured in wireless mmWave backhaul networks.

4.5.2.1. Full network case

Figure 4.6 shows the impact of the number of nodes and of the average data size on the makespan. Here we do not consider interference and we use $K=1$. As expected in this case, the makespan grows as long as the file size average grows, while heuristics reduce the makespan with respect to the direct download from the MBS. Indeed, `Resched` gets closer to the theoretical lower bound.

Due to the NP-hardness of the problem, optimum results are computationally unfeasible, thus we show optimality only for small networks, and with an average data size of 10 MB, in Figure 4.7. Here the optimum is computed by solving the MILP formulation for MMWBS. Bounds and heuristics are reported for comparison (and they are as in the slice obtained for average file sizes of 10 MB in Figure 4.6).

In Figure 4.7, the makespan grows linearly with the size of the network since the average burden of data at the MBS grows linearly with the number of nodes. We observe that the optimal results perform close to the lower bound and the heuristics, in particular, the `Resched` heuristic operates near-optimally when the MBS has one RF chain. `Greedy` achieves worse results than `Resched`, which is not surprising given its low computational complexity.

In Figure 4.8 we also show more optimal results for the makespan when the MBS disposes of $K = 2$ RF chains. Again, the makespan grows linearly with the size of the network, although the achieved values are much lower in this case. This fact is due to the possibility of using up to two simultaneous links from the MBS. In this case, `Resched` behaves better than `Greedy`, although the distance from the lower bound and from the optimum is higher than with $K = 1$. This is due to the fact that our heuristics give priority to direct download, when it can be fast, so that the MBS can transmit more often when K increases. Instead, we show the impact of K for the case in which we consider interference, in Figure 4.9. In there, we fix the size of the network to $n=15$ μBSs using antenna patterns radiating and interfering within $\pi/8$ rads. The figure shows that the makespan tends to decrease considerably as long as K increases. Such decrease follows an interesting shape that slows down the decrease until converging to an almost constant makespan. The reason behind this behaviour is that the more RF chains we have, the more parallel links can be active from the MBS and the less relaying takes place in the network. Still, `Resched` is able to reduce the makespan and take advantage of relaying in all the cases, despite the small gap of 50 ms (5 slots) between upper and lower bounds with high values of K. Indeed, the figure shows how tight our bounds are (although the lower bound is not necessarily a feasible schedule) and how `Resched` can achieve makespan reductions of 30% to 75% with respect to direct download (i.e., the upper bound) and a `Greedy` heuristic, using reasonable values of K. Note also that a greedy approach to relay yields no practical benefit as soon as the MBS can use two or three RF chains.

In general, Figures 4.7-4.9 show that `Resched` provides good results, but never

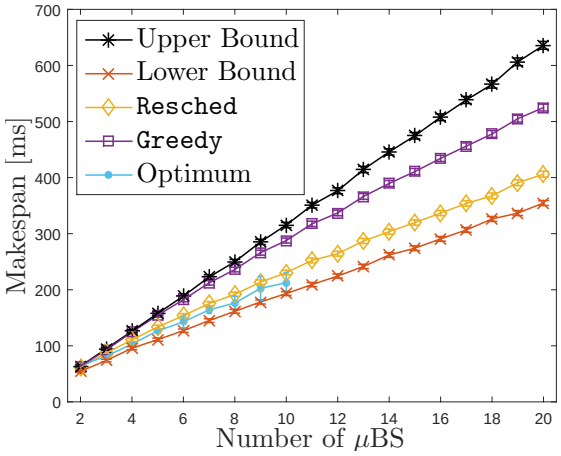


Figure 4.7: Full network case with $K = 1$, no interference, average data sizes of 10 MB and comparison to optimum.

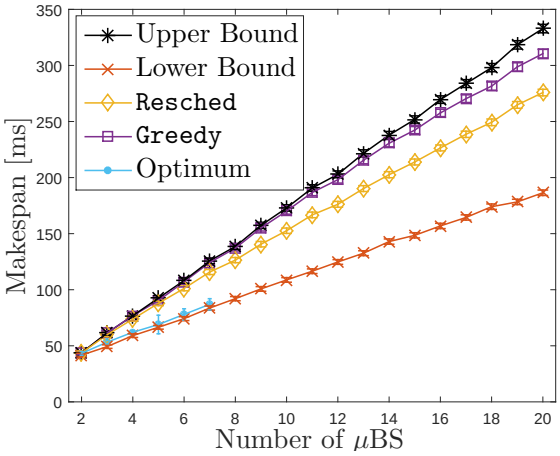


Figure 4.8: Full network case with $K = 2$, no interference: Impact of network size n, with comparison to optimum.

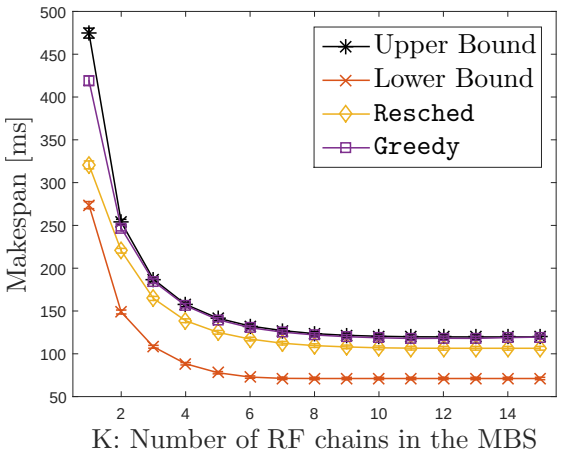


Figure 4.9: Full network case, with $n = 15$ and interference of HPBW $= \pi/8$ rads: Impact of K.

converges to the lower bound. Since heuristics always provide a makespan larger than the optimal, the optimal is not necessarily the lower bound. Heuristics aim to be as close as possible to optimal schedules, not to the lower bound. However, for those cases where obtaining the optimum schedule is not computationally feasible, `Resched` should approach as much as possible the lower bound, since it cannot be compared to other benchmarks.

4.5.2.2. Small cell network

We next take a set $\emptyset \neq R^R \subsetneq R$ of μBSs with no files ($d_r = 0, \forall r \in R^R$) whose only task is to help the network through relaying as detailed in Section 4.5.1. In this scenario, all other μBSs i in R^D do not relay traffic, which is implemented by setting to 0 the rate of

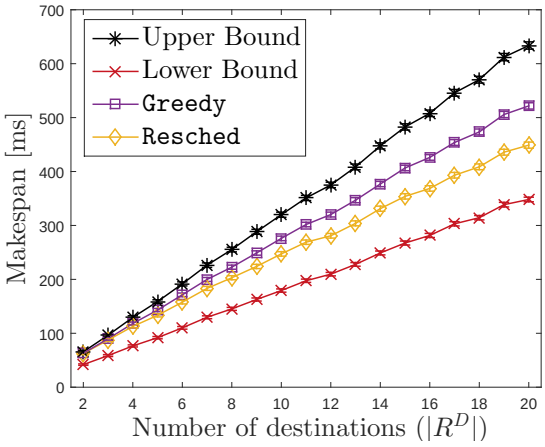


Figure 4.10: Small cell network case, with $K = 1$ and without interference: impact of the number of μBSs in R^D for a fixed number ($|R^R| = 15$) of relay μBSs in R^R.

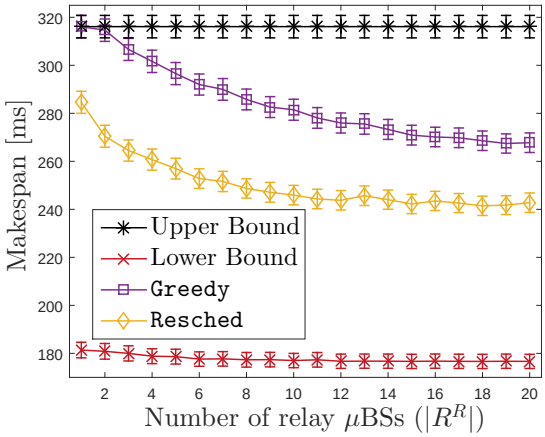


Figure 4.11: Small cell network case, with $K = 1$ and without interference: impact of the number of relay μBSs in R^R for $|R^D| = 10$ destinations.

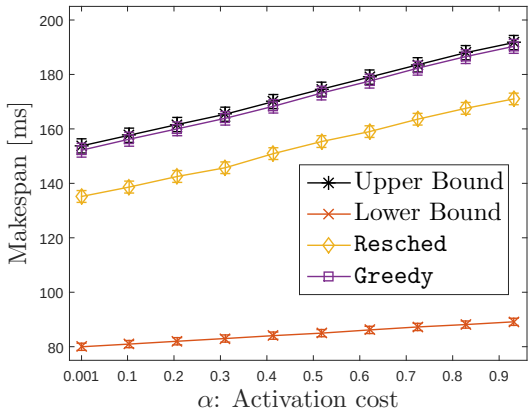


Figure 4.12: Small cell network case, with $K = 4$ RF chains in the MBS and with an interference of HPBW $= \pi/8$: impact of the activation cost, α, for $|R^R| = 15$.

any link (i, r). For a given small cell network, we consider a uniformly random placement of relays in R^R, as well as for other nodes.

In Figure 4.10 we show the impact of fixing the number of relay μBSs in R^R to 15 and increasing the destination nodes in R^D in the network. Here, without interference and with $K = 1$, the behaviour is expected to be a linear increase since we increase the traffic burden in the MBS as long as we add destination nodes. `Resched` behaves as a good scheduler for the makespan, not very far from the lower bound and reducing more than 50% the gap between both bounds, while `Greedy` suffers from the simplicity in its design.

More interestingly, to complement the results of Figure 4.10, in Figure 4.11 we show the impact of growing the small cell network through increasing the number of relays in R^R. We fix the number of end-nodes in R^D to $|R^D| = 10$. Again, we consider no

interference and assume $K=1$. We place randomly the destination nodes of R^D in each instance and keep increasing the number of relays. Such increase brings more chances for the end-nodes in R^D to be helped and then the makespan keeps reducing with both `Greedy` and `Resched`, although the gain flattens for high numbers of relays. The gain of `Resched` is remarkable, since it can practically save between 15% to 50% of time to complete the download of all data. Results with interference and with $K > 1$ yield very similar behaviours.

In Figure 4.12, we study the impact of the link activation cost, α. Since the behaviour is similar in all cases, we only show an example with $K=4$ and with interference. We find that the makespan of heuristics and of bounds increases linearly with the activation cost. The slope of the lower bound is, however, smaller than for the other curves. This result is due to the fact that our heuristics and the upper bound do not allow to split a data download in multiple separate chunks. Instead, when a download is scheduled, the entire file is transmitted. The increase of the lower bound is less remarkable here because such lower bound is computed with the LP of Figure 4.4, which does not use time slots, so that it can be more efficient (although the resulting scheduling is not necessarily feasible). In practice, while increasing the link activation cost may provoke the increase of an integer number of time slots for heuristics and upper bound, the LP used for the lower bound only increases file transmission times by α, without using any discretized schedule. Here we have also tested an activation cost of 1 μs, for those cases in which one can assume that the beam-training is saved an not repeated when links activate.

4.5.2.3. Interference and spectrum reuse

In Figure 4.13 we study the impact on the makespan in the presence of interference, as a function of the HPBW, for a full network of 10 μBSs and $K = 2$. We observe how beamwidths below $\pi/4$ rads barely affect network performance, and so `Resched` behaves in a similar way in presence or absence of interference. However, as interference increases because of larger beamwidths, `Resched` becomes less impaired than `Greedy`. Still, the overall makespan increases because the interference limits network capacity. For beamwidths of π rads the performance of the makespan is already like the direct downloads without spectrum reuse. We also compare the behavior of `Resched` and `Greedy` by considering the reuse of links in the presence of interference, as shown in Figure 4.14 and Figure 4.15. Clearly, `Resched` achieves higher reuse factors, even though the degree of interference considered in the figure is low (with beamwidths of $\pi/8$ rads). Specifically, in Figure 4.14 we use *box-and-whisker* plots to observe the distribution for the ratio of time slots in which a number of links are active when we have $K=4$ RF chains in the MBS. Figure 4.15 reports just average values for such ratio, and compares different values of K. Since there can be up to K simultaneous transmissions from the MBS, this is often the most common number of simultaneous links active in the network. However, we can

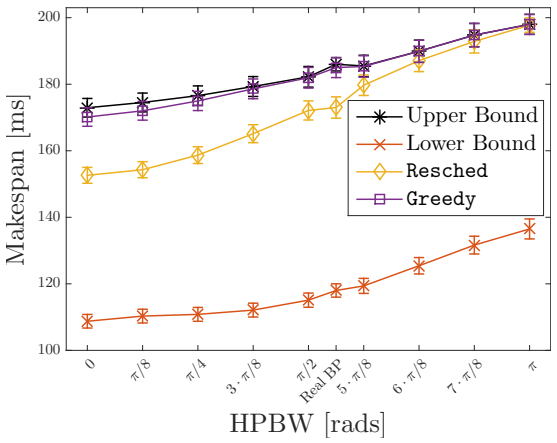


Figure 4.13: Full network case, with $n = 10$ μBSs and $K = 2$: impact of non-ideal beamwidths causing an interference.

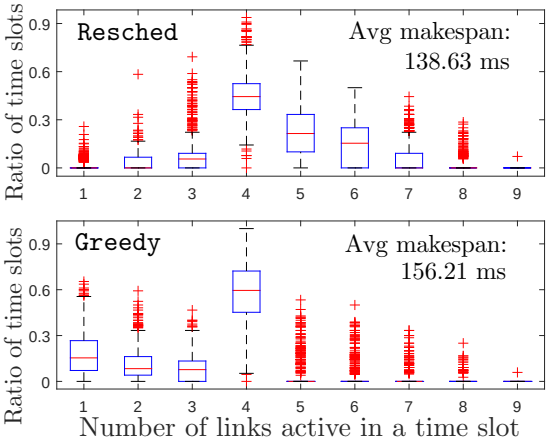


Figure 4.14: Full network case, with $n = 15$ μBSs and $K = 4$: reuse of links with interference caused by a HPBW of $\pi/8$ rads.

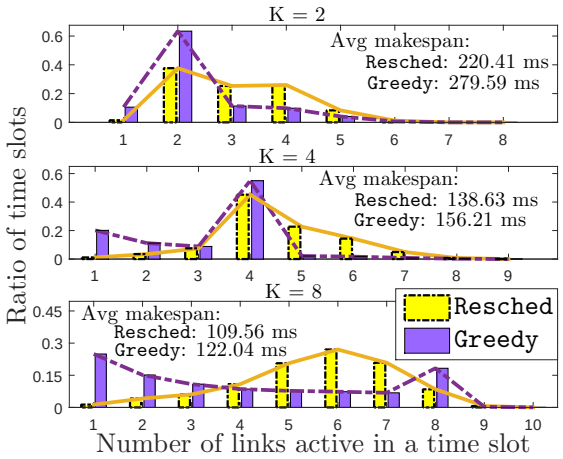


Figure 4.15: Full network case, with $n = 15$ μBSs and $K = 2, 4, 8$ RF chains in the MBS. Distribution of spatial reuse with interference caused by HPBW $= \pi/8$ rads.

see from the figures that `Resched` is able to take advantage of relaying and can use even more than K active links per time slots with high frequency, while `Greedy` suffers from its simplicity and often uses K active links. Nonetheless, we note from the bottom part of Figure 4.15 that `Resched` uses less than K links in parallel with high frequency when K is high. The resulting makespan is however shorter than for `Greedy`. The reason behind this counter-intuitive example is that high spatial reuse does not necessarily lead to faster downloads when interference can build up, which takes us to the next set of results, in which we consider the rates actually used.

Figures 4.16 and 4.17 depict the distribution of aggregate network rates for `Resched` and `Greedy`, respectively. In each of the two figures we report the distribution with

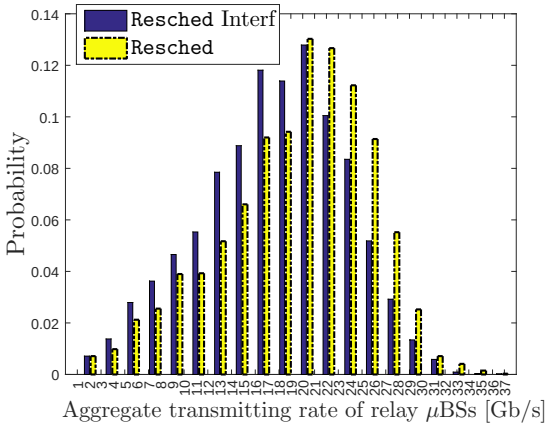


Figure 4.16: Full network case, with $n = 15$ μBSs and $K = 8$: distribution of aggregate rate of μBSs with `Resched` without and with interference of $\pi/8$ rads HPBW.

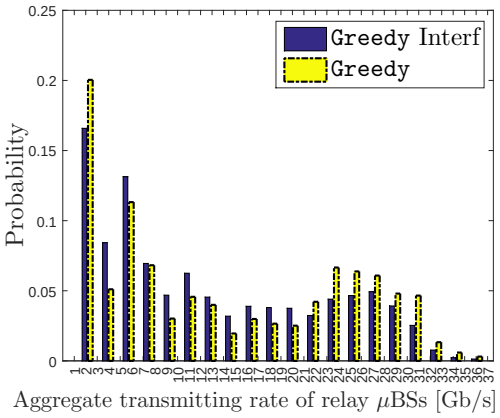


Figure 4.17: Full network case, with $n = 15$ μBSs and $K = 8$: distribution of aggregate rate of μBSs with `Greedy` without and with interference of $\pi/8$ rads HPBW.

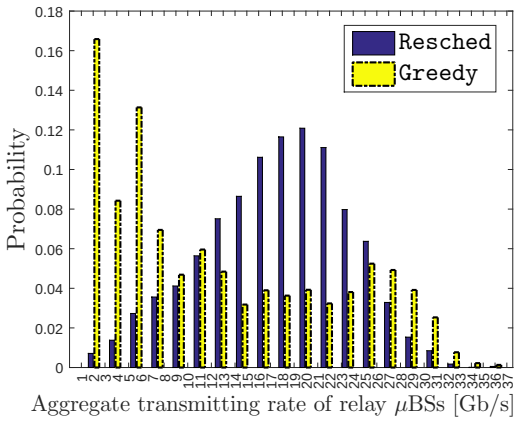


Figure 4.18: Full network case, with $n = 15$ μBSs, $K = 8$ and interference of HPBW $= \pi/8$ rads: distribution of aggregate rate of μBSs with `Resched` and `Greedy`.

and without interference, for $K = 8$ and a full network with $n = 15$. We observe that the presence of interference tends to reduce the use of better links. Finally, Figure 4.18 provides a direct comparison between the two heuristics in the presence of interference. Concretely, with beam-patterns having a HPBW of $\pi/8$ rads, 15.6% of the pairs of links interfere, on average. Here, although `Greedy` selects always the fastest available links among relays, it ends up providing worse aggregate rates and longer makespans than `Resched` because it forces the use of direct downloads in absence of good inter-μBS links. Instead, `Resched` avoids scheduling too many links when interference builds up. Therefore `Resched` achieves high spatial reuse without penalizing speeds.

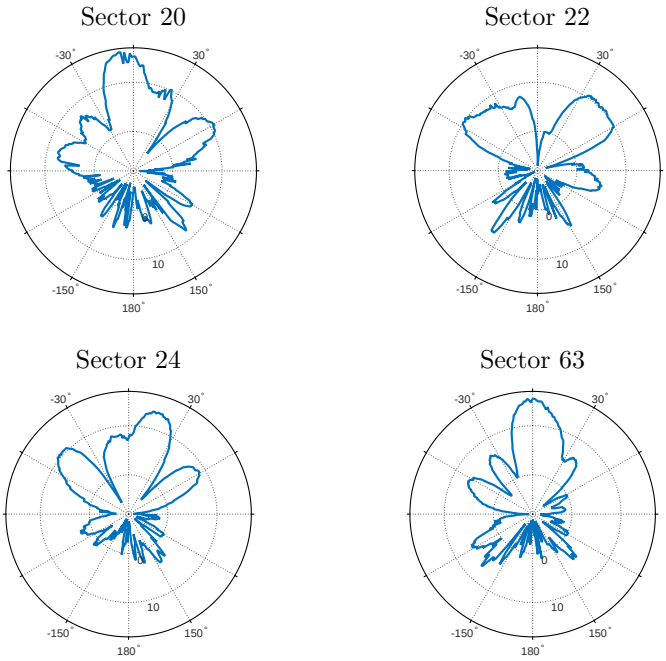


Figure 4.19: Measured beam-patterns in [114] from commercial off-the-shelf mmWave devices.

4.5.2.4. Real-measured interference and link rates

Finally, we provide some results from scenarios with realistic measured beam-patterns and data rates from mmWave devices. Here, the interference map is based on real shapes of beam-patterns for cheap commercial antennas [114, 115] and the makespan directly depends on real rates observed in 802.11-based mmWave devices [116]. In Figure 4.19 we show four of the 35 available beam-patterns integrated on the TP-Link Talon AD7200. Here we can observe that although communication is directional, there are relevant sidelobes that incur strong interference in the system. Hence, specific algorithms that carefully manage interference in order to provide high spatial reuse are needed, as `Resched` or `Greedy`. In Figure 4.20 we show the spatial reuse when interference is caused by real beam-patterns. As observed before, `Resched` provides higher spatial reuse than `Greedy` and hence provides a lower average makespan. In comparison with Figure 4.15, we observe a lower spatial reuse in Figure 4.20 because there is more presence of non-negligible interference that affects the possibilities of spatial reuse. This fact also leads to lower aggregate transmit rate values in Figure 4.21, where we observe better spatial reuse in which aggregate rates are higher, but lower than the ones observed in Figure 4.18. Moreover, real SNR measures obtained in [116] are based on the imperfect beam-pattern shapes shown and hence provide lower link rates. In this case, 27.4% of the pairs of links interfere, on average.

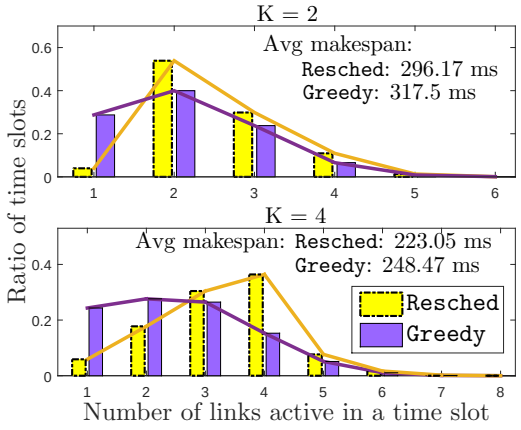


Figure 4.20: Full network case, with $n = 15$ μBSs and $K = 2, 4$ RF chains in the MBS. Distribution of spatial reuse with interference caused by real beam-patterns measured in [114].

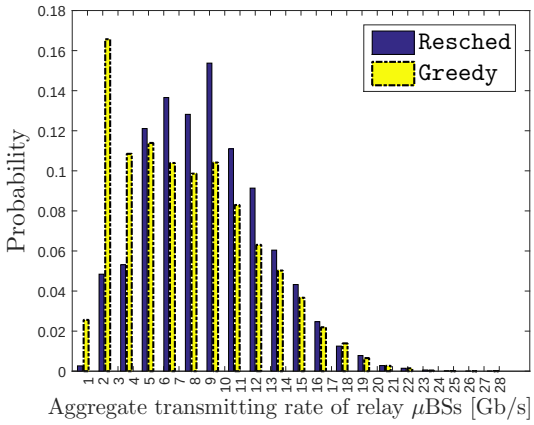


Figure 4.21: Full network case, with $n = 15$ μBSs, $K = 8$ and interference caused by real beam-patterns [114]: distribution of the aggregate rate of μBSs.

4.6. Lessons Learnt and Discussion

Our analysis has shown that, in general, minimizing the delivery time (i.e., the makespan) of a collection of files in a mmWave backhaul network is not doable in polynomial time, unless $P = NP$. Even in simple scenarios where mmWave beams are fine-grained enough so that interference is neglected and the MBS only has one RF chain to transmit, the problem is NP-hard. So, time-efficient heuristics as `Resched` and `Greedy` are necessary to find feasible solutions. Our results show that our heuristics, specially `Resched`, are able to provide near-optimal solutions that are, with respect to the distance between upper and lower bounds, 40-80% closer to the lower bound. Hence, since optimal scheduling cannot be implemented due to time constraints, one should always implement the `Resched` heuristic, which approximates the optimal better than `Greedy`. However, the complexity of `Resched` is quadratic with the network size, while `Greedy`'s is linear. Hence, only in those cases in which the time of the decision-making process is really tight, one would implement `Greedy`. Furthermore, for the few cases in which optimal solutions are computationally feasible, we have observed that relay reduces the makespan by a significant 35%. The results shown in Figures 4.7-4.12 illustrate how enabling relay considerably mitigates the transmission bottleneck at the MBS. Indeed, as shown in Figures 4.13 to 4.21, using relay is convenient with ideal and realistic antenna patterns, since it allows efficient spatial reuse with high probability to achieve high aggregate rates.

As we show in Figure 4.22 for a typical case, direct download from the MBS only accounts for less than half of the aggregate utilization of links in the network. File download from the MBS to μBS relays occupies basically the same as μBS-to-μBS links,

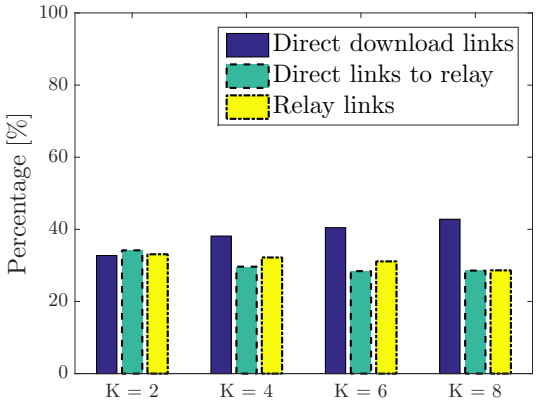


Figure 4.22: Full network case, with $n = 15$ μBSs and interference of $\pi/8$ rads HPBW. Usage of direct and relay links.

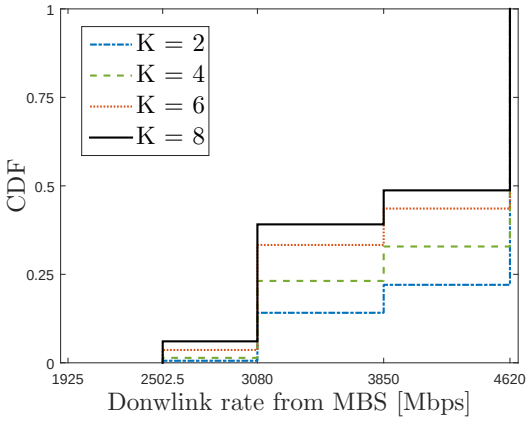


Figure 4.23: Full network case, with $n = 15$ μBSs and interference of $\pi/8$ rads HPBW. CDF of downlink rates from the MBS whose links are used to relay traffic.

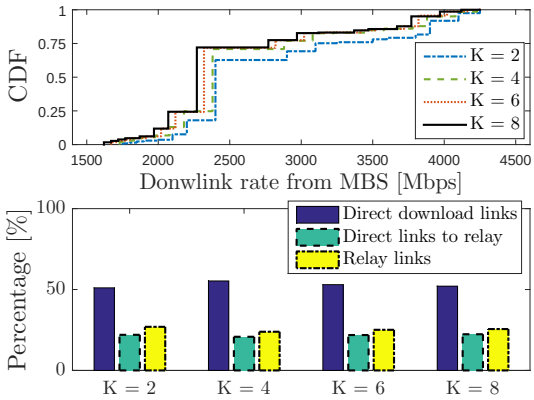


Figure 4.24: Full network case, with $n = 15$ μBSs. Interference is as measured in [114] and rates depend on distance [116]. CDF of downlink rates from the MBS whose links are used to relay traffic (top); and usage of direct and relay links (bottom).

which indicates not only that relaying is convenient but also that our schemes do not blindly offload files to relays with inadequate connectivity quality to the destination μBS. `Resched` in fact tends to use relays that require a similar time to receive and retransmit the selected files. To provide some insights on which μBSs are usually selected and why, we show in Figure 4.23 the CDF of downlink rates of those μBSs that act as relays. The CDF has a staircase shape because only discrete values of rates are possible, each corresponding to a given MCS. We observe in the figure that only μBSs with the best three MCS values are selected as relays, and more than 50% of relays use the maximum MCS.

The gain shown for our optimization can be impaired by non-ideal beam-patterns. For instance, for the extreme case of real beam-patterns of cheap antennas as the ones

studied in Figures 4.19 to 4.21, we show in Figure 4.24 that the CDF of relay rates is more spread and exhibit poorer statistics. For instance, the median for the full network case with 15 μBSs and $K=8$ is 2.8 Gbps, against the 3.8 Gbps of the case of ideal antennas with HPBW=$\pi/8$ rads. However, the gain remains considerable as well as the fact that our optimization leads to dedicate about half of the link usage to relay, and to balance relay's in and out traffic, like in the case of ideal antennas. As a remark, while the use of cheap antennas could be common for 802.11-based inexpensive indoor devices, it is less reasonable to mount them on towers and outdoor deployments covering relatively large areas, which is where relay might be needed the more.

As a result of the lessons learnt on this research, we answer the questions raised at the introduction about the high complexity of the whole framework, the convenience of relaying and spatial reuse, and the selection of proper relays to know which μBSs relay traffic to which μBSs.

Part II

Dynamic Relay Optimization

In this part, we study relay methods that leverage emerging extremely mobile paradigms such as aerial relay. Here, we envision aerial-assisted cellular networks in which it is desirable that traffic of users follows relayed paths from ground base stations through aerial relays mounted on drones. As the aerial space offers a vast amount of possibilities to (re-)position aerial relays, the main goal in this part is to derive optimal aerial placement of relays to guarantee best network performance in terms of guaranteed Quality-of-Service (QoS) coverage and fair user throughput. We model backhaul and backbone capacity constraints, interference and multiple ground cells jointly coordinated with a fleet of drone relays.

5 Coverage Optimization with a Dynamic Fleet of Drone Relays

In this chapter we focus on the optimization of 3–D hovering positions and flight routes for a fleet of drone relays aiding a ground cellular network, as depicted in Figure 5.1. Drones are coordinated yet they mutually interfere. We optimize coverage based on the QoS offered by drones under realistic path-loss models for Line-of-Sight (LoS) and Non-LoS (NLoS) communications and interference. Considering interference is key because it results in radically different coverage footprints and imposes strict constraints on the position of drones with respect to the position of ground base stations. We use *Extremal-Optimization* (EO) [118] and propose the On-demand Drone Coverage (`OnDrone`) algorithm, an *extremal-optimization* algorithm that computes near-optimally joint positions for drones, based on realistic assumptions on previous drone positions and interference, which is otherwise an intractable NP-Complete problem. We also propose for the first time the use of *Bézier curves* [79] for flight routes aiming to enhance communications over time.

We assess the benefits of our optimization framework by (*i*) comparing `OnDrone` against the optimal solutions and state-of-the-art approaches for tractable cases; (*ii*) performing numerical simulations for larger networks with realistic topologies and environmental constraints; and (*iii*) evaluating fleet repositioning using either *Bézier curves* or straight paths as drone routes. Our numerical results show that `OnDrone` is nearly optimal and outperforms state-of-the-art coverage solutions as proposed in [119] and [74]. Also, we show that the use of *Bézier curves* is key to boost coverage when studying drone repositioning in dense urban scenarios, and shows remarkable advantages over straight paths, as adopted in [78].

The main contributions of this chapter are summarized:

- We propose a dynamic drone relay-aided network in which we maximize the coverage of ground users by means of aerial base stations with an interference-aware on-demand multi-drone coverage framework that accounts for both user access and backhaul links.

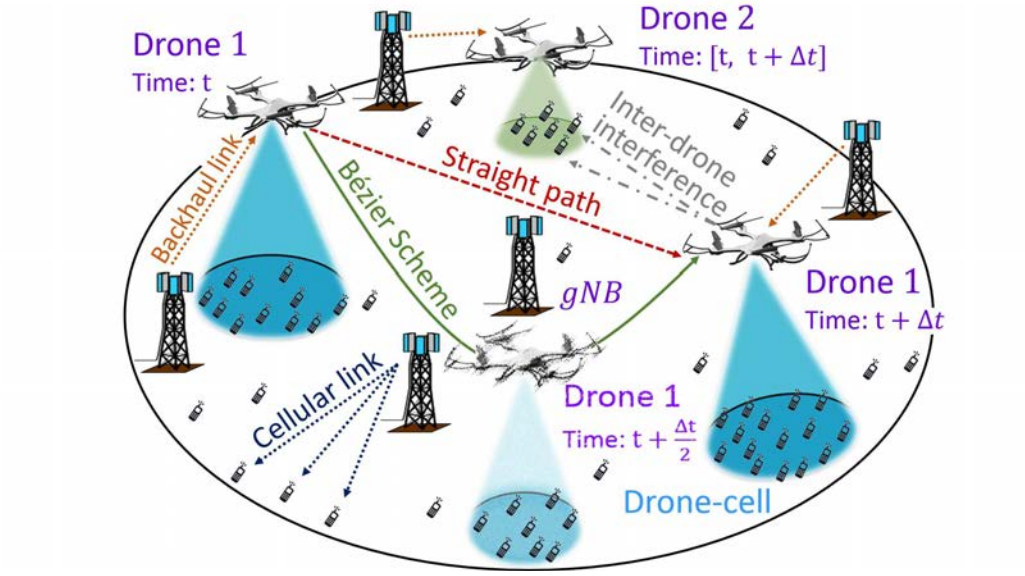


Figure 5.1: Reference scenario: multi-drone-aided network.

- We prove that the problem is NP-Complete.
- We propose `OnDrone`, a light-weight algorithm based on *extremal-optimization* that solves the problem *on-demand*.
- We propose the use of a strong geometrical tool to design the flight paths of drones: the *Bézier Scheme*.
- We assess our proposals in realistic scenarios and topologies in comparison with state-of-the art solutions and show the gain of our proposals.

The rest of the chapter is structured as follows. Section 5.1 presents the system model assumptions for the reference scenario and wireless channels, while Section 5.2 states and formulates the coverage problem, and shows that it is intractable. Section 5.3 details the overall optimization framework. Section 5.4 reports numerical results. Section 5.5 presents a discussion on the lessons learnt in this chapter.

5.1. System Model

5.1.1. Reference Scenario

We consider a ground surface $\mathcal{S}$ administrated by the ground network consisting of a set $\mathcal{G}$ of ground base stations, as shown in Figure 5.1.[1] We refer to ground base stations as Next Generation Nodes B (*gNB*s), using the new 3rd Generation Partnership

[1]New Software Defined Network (SDN) tools designed to manage large networks are able to coordinate ground base stations to perform any network optimization [9].

Project (3GPP) jargon for next generation Base Stations (BSs). In the region $\mathcal{S}$, the ground network provides service to a set $\mathcal{U}$ of User Equipments (UEs), i.e., mobile users. In order to increase coverage, we consider that coverage assistance is provided by the presence of a finite set $\mathcal{D}$ consisting of D drone relay stations. Each drone is equipped with a mobile relay that gives access to UEs on an orthogonal downlink bandwidth with respect to the *gNB*s band. We refer to drones as aerial Base Stations (*aBS*s).

We assume that *gNB*s provide backhaul connectivity to *aBS*s over the reuse of the downlink spectrum used for *gNB*–UEs access links. Current *gNB*s provide cellular coverage through three sectors pointing mainly to the ground. We assume that in order to set backhaul *gNB*–*aBS* links, *gNB*s have an additional full dimensional antenna array that performs 3D–beamforming over clear LoS links[2], as suggested and studied in [120]. Therefore, access links *gNB*–UEs and backhaul links *gNB*–*aBS* do not practically interfere. Furthermore, *gNB*s equipped with this kind of antenna array are able to perform 3D–beamforming to several relays, and alternate transmissions over time slots on a millisecond scale. Hence, each *gNB* g can provide backhaul service to a limited number of *aBS*s, namely D_g.

The coverage of each *aBS* is an irregular ground area that depends on the drone height, cell environment and interference from other *aBS*s. The interference among *aBS*s directly affects the Signal-to-Interference-plus-Noise Ratio (SINR) that the ground users receive. The SINR depends on the air-to-ground path-loss model, which is based on the link LoS probability between drones and users. As described later, such path-loss model clearly differs from the conventional attenuation models used in ground cellular networks. Indeed, such a LoS-based path-loss and interference model for the communications channel provides a multi-*aBS* coverage framework for aerial networks, which is radically different from conventional frameworks for ground networks—as e.g., ground Device-to-Device (D2D) networks like shown in Chapter 3—and whose characteristics we study. In this framework, we consider that a *gNB* g and an *aBS* d can realistically serve a limited number of users, namely U_g and U_d, respectively. We further assume that channel bandwidth is equally split among the users that a BS serves, although more sophisticated schedulers could be easily adopted in the analysis.

With the above, we aim to find optimal locations for D drones, so as to maximize the number of users covered by *gNB*s *and aBS*s with a guaranteed bandwidth. Besides, we identify two additional problems to support fleet repositioning: (i) deciding which drone flies to which position upon an optimization update and (ii) designing flight routes.

[2]Based on the receiver location or instantaneous channel state information, 3D–beamforming allows to build directional beam-patterns that focus the transmission energy on the direction where the receiver is. This flexible technique helps to mitigate interference so as to provide higher rates. 3D–beamforming is very useful for backhaul wireless links from one source to few relays, as in an aerial backhaul network.

5.1.2. Air & Ground Channels: Path-Loss and Interference

We assume that the surface $\mathcal{S}$ is flat, so that the position of a UE $u \in \mathcal{U}$ is taken as an input and denoted by $\pi^u = (x^u, y^u, 0)$. The position of a *gNB* $g \in \mathcal{G}$ is known—as this is public information—and denoted by $\Pi_g = (X_g, Y_g, h_g)$. The positions of all *aBS*s $d \in \mathcal{D}$ are the decision variables of the coverage problem, and denoted by $\Pi^d = (X^d, Y^d, h^d)$.

5.1.2.1. Air-to-ground access channel

For all drone $d \in \mathcal{D}$, and for all user $u \in \mathcal{U}$, the horizontal distance between u and the ground projection of d is $r_{d,u} = \|(X^d, Y^d) - (x^u, y^u)\|$. The elevation of d is h^d. Due to the low altitude of drones—a few hundreds of meters at most—the channel conditions of communications between a serving drone and an end-user are much affected by the LoS. Depending on whether the access link is free of obstacles (like buildings, traffic, etc.), the attenuation differs considerably [121]. Thus, the air-to-ground path-loss among *aBS*s and UEs depends on the probability of LoS, which is a complex function of the elevation angle between user u and drone d, according to the following expression:

$$P_{LoS}(d,u) = \frac{1}{1 + a \cdot e^{-b\left(\frac{180}{\pi}\arctan\left(\frac{h^d}{r_{d,u}}\right) - a\right)}}, \tag{5.1}$$

where a, b are parameters depending on the environment, i.e., number of buildings and large signal obstructions per unit area, building's height distribution, ratio of built-up area and clean surfaces, etc., as it has been derived in [67], based on the ITU recommendations [122]. In Eq. (5.1), the elevation angle (in radians) appears as $\theta_{d,u} = \arctan(h^d / r_{d,u})$. As $\theta_{d,u}$ approaches $\frac{\pi}{2}$, i.e., when the drone d hovers just above the user u, the probability of LoS reaches its maximum value. The elevation angle $\theta_{a,u}$ is fully characterized by the *aBS* height h^d and the ground distance between the user and the *aBS*, $r_{d,u}$.

In Figure 5.2, we see that the positions of the drones directly affect blockage conditions of the *aBS*–UE access links. Thus, the higher a drone hovers, the more likely is to have LoS. However, the strength of the signal gets also attenuated with the distance. For single-drone missions, there is an optimal altitude that maximizes coverage [67]. However, in a multi-drone scenario as the one we discuss in this chapter, the effects on interference from other drones are a key additional issue to consider, one that makes the optimal drone hovering altitude depend on the positions and elevations of the rest of the drones. This also precludes the possibility to straightforwardly apply single-drone mission approaches to multi-drone scenarios, since the former are not designed to account for interference, as in [119].

While ground cellular links suffer from conventional attenuation based on fast and

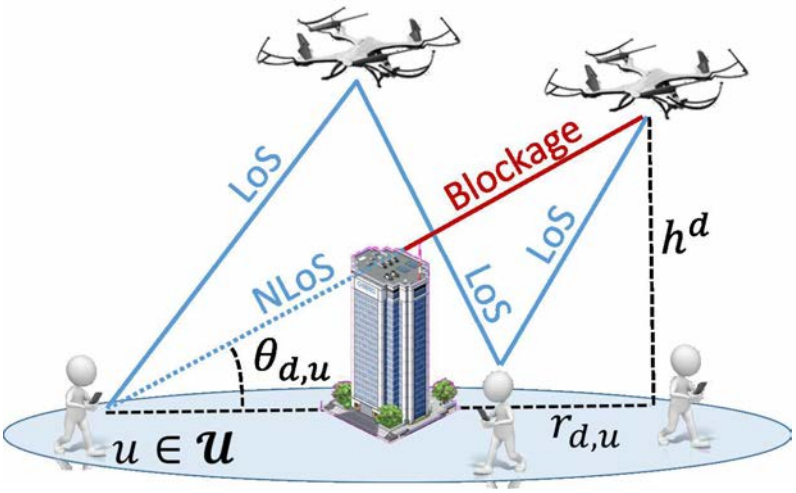


Figure 5.2: Reference illustration of LoS conditions.

slow fading, as detailed later in this section, the path-loss of an *aBS*–UE link (d, u) differs notably and is affected by an *excess attenuation*, depending on the LoS likelihood presented in Eq. (5.1). The average attenuation is derived in [67] as:

$$\begin{aligned} \mathcal{L}_{\mathcal{A}}(d,u) &= 10\eta_{\mathcal{A}} \log_{10}\left(\frac{4\pi f_{\mathcal{A}}}{c_l}\cdot\sqrt{(h^d)^2 + r_{d,u}^2}\right) \\ &+ P_{LoS}(d,u)\cdot(\xi_{LoS} - \xi_{NLoS}) + \xi_{NLoS}, \end{aligned} \tag{5.2}$$

where ξ_{LoS}, ξ_{NLoS} are the *excess path-loss* components in LoS and NLoS connections respectively, $\eta_{\mathcal{A}} = 2$ is the path-loss exponent and $f_{\mathcal{A}}$ is the carrier frequency in Hz. As surveyed in [123], we have based the air-to-ground path-loss on the average large-scale fading in order to perform the *aBS*–UE association in the optimization. Nevertheless, we further consider fast and slow fading, modelled as log-normal and Rayleigh distributions, respectively, in the implementation of the framework when we analyze the results. We denote the access link SINR as $\Gamma_{d,u}^{\mathcal{A}}$, which is equal to:

$$\Gamma_{d,u}^{\mathcal{A}} = \frac{P_{Tx}^{d} \cdot 10^{-\mathcal{L}_{\mathcal{A}}(d,u)/10}}{N_u + I_{d,u}^{\mathcal{A}}}, \tag{5.3}$$

where P_{Tx}^{d} is the transmission power of the antenna integrated in the *aBS* d, N_u is the thermal noise, and most importantly, $I_{d,u}^{\mathcal{A}}$ is the actual interference that user u suffers from any other *aBS*s. Thus, the interference depends on the 3–D position of the rest of the *aBS*s, i.e.:

$$I_{d,u}^{\mathcal{A}} = \sum_{d' \in \mathcal{D}\setminus\{d\}} P_{Tx}^{d'} \cdot 10^{-\mathcal{L}_{\mathcal{A}}(d',u)/10}. \tag{5.4}$$

Table 5.1: Channel modelling

Channel	***Path-loss***	***Exponent***
Ground-to-Ground (*gNB*–UE)	$10\eta_G \log_{10}\left(\frac{4\pi f_G}{c_l}\right) + 10\eta_G \log_{10}\left(\|\Pi_g - \pi^u\|\right) + \mathcal{N}_{\sigma_G}$	$\eta_G > 2$
Ground-to-Air (*gNB*–*aBS*)	$10\eta_{\mathcal{B}} \log_{10}\left(\frac{4\pi f_{\mathcal{B}}}{c_l}\right) + 10\eta_{\mathcal{B}} \log_{10}\left(\left\|\Pi_g - \Pi^d\right\|\right) + \mathcal{N}_{\sigma_{\mathcal{B}}}$	$\eta_{\mathcal{B}} \approx 2$
Air-to-Ground (*aBS*–UE)	$10\eta_{\mathcal{A}} \log_{10}\left(\frac{4\pi f_{\mathcal{A}}}{c_l}\left\|\Pi^d - \pi^u\right\|\right) + P_{LoS}(d,u)\cdot(\xi_{LoS} - \xi_{NLoS}) + \xi_{NLoS}\ (+\mathcal{N}_{\sigma_{\mathcal{A}}} + \mathcal{R}_{\varsigma_{\mathcal{A}}})$	$\eta_{\mathcal{A}} = 2$

We impose that the minimum rate, $\frac{1}{U_d} W_{\mathcal{A}} \log_2(1+\Gamma^{\mathcal{A}}_{d,u})$, is above $R^{\mathcal{A}}_{\min}$, where $W_{\mathcal{A}}$ is the access channel bandwidth.

In our system model, *aBS*s operate in orthogonal bandwidth with the cellular band. Thus, there is no interference between cellular users and drone-served users, which is commonly the main limiting factor in aided cellular networks, as e.g., *inband* D2D networks like shown in Chapter 3.

5.1.2.2. Ground-to-ground access channel

Ground cellular links, i.e., *gNB*–UE access links, operate over an OFMDA channel with access bandwidth $W_{\mathcal{A}}$. Hence, users scheduled by the same *gNB* do not suffer intra-cell interference. However, ground users may enjoy poor QoS due to the presence of inter-cell interference, from other close *gNB*s. The path-loss of these channels is based on large- and small-scale fading, as widely studied in literature [124] and shown in Table 5.1. We denote the SINR of ground access links (g,u) as $\Gamma^{\mathcal{A}}_{g,u}$ and impose that its minimum user access rate, i.e., $\frac{1}{U_g} W_{\mathcal{A}} \log_2(1+\Gamma^{\mathcal{A}}_{g,u})$, is above the guaranteed rate $R^{\mathcal{A}}_{\min}$.

5.1.2.3. Ground-to-air backhaul channel

In order to provide backhaul connectivity to *aBS*s, *gNB*s perform 3D–beamforming over clear LoS links. Hence, the attenuation that a signal from *gNB* g to *aBS* d suffers is:

$$\mathcal{L}_{\mathcal{B}}(g,d) = 10\eta_{\mathcal{B}} \log_{10}\left(\frac{4\pi f_{\mathcal{B}}}{c_l} \cdot \left\|\Pi_g - \Pi^d\right\|\right) + \mathcal{N}_{\sigma_{\mathcal{B}}}, \tag{5.5}$$

where $\eta_{\mathcal{B}} \approx 2$ is the path-loss exponent in free-space LoS links, $f_{\mathcal{B}}$ is the frequency of backhaul wireless links in Hz, c_l is the speed of light in m/s and $\mathcal{N}_{\sigma_{\mathcal{B}}}$ is a random gaussian variable with zero mean and $\sigma_{\mathcal{B}}$ standard deviation, modelling the effects of slow fading and shadowing.

Table 5.2: Channel interference

	Ground-to--Ground	**Ground--to-Air**	**Air-to--Ground**
Ground-to--Ground	Inter-cell interference	Directional re-use	Orthogonal bands
Ground--to-Air	Directional re-use	Low interference: 3D–beamforming	Orthogonal bands
Air-to--Ground	Orthogonal bands	Orthogonal bands	Inter-drone interference

3D–beamforming builds antenna patterns that radiate much of the energy over a main lobe with a Half-Power Beamwidth (HPBW) that may be wide, hence incurring high interference to other *aBS*s in LoS. Also, the formation of directional beam-patterns comes with the presence of side-lobes with non-negligible radiating power, that also incur (low) interference. Thus, depending on the radiating angle of other *gNB*s, a backhaul link may enjoy better or worse QoS, due to the presence of interference[3]. We denote the backhaul SINR of link (g, d) as $\Gamma^{\mathcal{B}}_{g,d}$, which is equal to:

$$\Gamma^{\mathcal{B}}_{g,d} = \frac{P^{g}_{Tx} \cdot G_g \cdot 10^{-\mathcal{L}_{\mathcal{B}}(g,d)/10}}{N^d + I^{\mathcal{B}}_{g,d}}, \tag{5.6}$$

where P^{g}_{Tx} is the transmission power of g, G_g is the antenna gain over the main lobe of the beam-pattern of g, N^d is the thermal noise, and most importantly, $I^{\mathcal{B}}_{g,d}$ is the actual interference that *aBS* d suffers from any other *gNB*, depending on the angle of their beam-patterns, i.e.:

$$I^{\mathcal{B}}_{g,d} = \sum_{g' \in \mathcal{G} \setminus \{g\}} P^{g'}_{Tx} \cdot G_{g'}(\phi_{g',d}) \cdot 10^{-\mathcal{L}_{\mathcal{B}}(g',d)/10}, \tag{5.7}$$

where $\phi_{g',d}$ is the angle between the direction of the main lobe of the antenna of g' and the position of *aBS* d. We impose that the minimum backhaul rate, $\frac{1}{D_g} W_{\mathcal{B}} \log_2(1+\Gamma^{\mathcal{B}}_{g,d})$, is above a rate $R^{\mathcal{B}}_{\min}$. $W_{\mathcal{B}}$ is the backhaul channel bandwidth.

In Table 5.1 we gather the path-loss model used for each kind of channel, where f_G is the band used for *gNB*–UE access links and is equal to the *gNB*–*aBS* backhaul links, i.e., $f_G = f_{\mathcal{A}}$, and $\mathcal{R}_{\varsigma_{\mathcal{A}}}$ is a random Rayleigh variable with scale parameter $\varsigma_{\mathcal{A}}$. In Table 5.2 we summarize the interference suffered in each of the channels, as it has been described along this section. We have shadowed the table cells that imply presence of interference.

[3]In general, since beamforming builds antenna patterns with directional main lobes, interference remains low for non-aligned *aBS*s

5.2. Multi-Drone Coverage Framework

We aim to find optimal 3–D positions for a fleet of D drones in which the number of UEs under network coverage is maximum. The optimization is run at regular time intervals, considering every time the users as static, so that static drone positions solve the coverage problem. The coverage maximization provides a set of 3–D coordinates where drones have to fly during the time interval, provided a drone d has to fly towards a reachable destination, i.e., a point within the ball $\mathcal{S}_d$ of radius given by the drone speed times the duration of the optimization update interval. However, the output of the optimization does not necessarily coincide with the assignment of fleet destinations that also minimizes flight time. Hence, we will solve the assignment of fleet destinations as a secondary problem, given the optimal coordinates found by the primary problem.

Although it is possible to formulate the coverage maximization and the minimum flight time assignment in one optimization problem, we believe that it is more clear to decouple both problems and solve them separately, while having the same optimal solution. This is due to the fact that the set of optimal drone positions is not changed by solving the secondary problem and at least one feasible solution exists for the secondary problem, which is the output of the primary problem. Thus, we first present the optimal aerial coverage (Section 5.2.1) and then the assignment of fleet destinations (Section 5.2.2).

5.2.1. Optimal Aerial Coverage

Coverage Problem $\mathcal{C}$: *Given a fleet $\mathcal{D}$ of drone relays in a cellular network managed by a centralized orchestrator, U ground users, a height range $[h_{\min}, h_{\max}]$ for the aBSs, guaranteed coverage rates $R^{\mathcal{A}}_{\min}$, $R^{\mathcal{B}}_{\min}$ for access and backhaul channels respectively, and a maximum number of users U_g and U_d that gNBs and aBSs can cover, find the optimal 3–D positions $\Pi^d = (X^d, Y^d, h^d)$ of drones so as to maximize the amount of users covered by ground- and drone-cells.*

Since the positions of drones, including their heights, affect the shape of the covered regions, we can mathematically formulate the Coverage Problem $\mathcal{C}$ to search for optimal values of the continuous decision variables $\Pi^d = (X^d, Y^d, h^d) \in \mathbb{R}^3$ that maximize the number of users under network coverage. Denoting by $C_{b,u}$ the binary variable that takes value 1 if BS $b \in \mathcal{G} \cup \mathcal{A}$ covers user u and 0 otherwise, and by $B_{g,d}$ the binary variable that takes value 1 if *gNB* g provides backhaul service to drone d and 0 otherwise, the formulation of the Coverage Problem $\mathcal{C}$ is:

$$
\begin{cases}
\max\limits_{\{\Pi^d\}_{d\in\mathcal{D}}} \ \sum\limits_{u\in\mathcal{U}}\left(\sum\limits_{g\in\mathcal{G}} C_{g,u}+\sum\limits_{d\in\mathcal{D}} C_{d,u}\right); \\
\text{subject to:} \\
\textit{Access network constraints:} \\
\frac{1}{U_b}W_{\mathcal{A}}\log_2\left(1+\Gamma^{\mathcal{A}}_{b,u}\right)\geq R^{\mathcal{A}}_{\min}\cdot C_{b,u}, & \forall b\in\mathcal{G}\cup\mathcal{D},\forall u\in\mathcal{U}; \\
\sum\limits_{g\in\mathcal{G}} C_{g,u}+\sum\limits_{d\in\mathcal{D}} C_{d,u}\leq 1, & \forall u\in\mathcal{U}; \\
\sum\limits_{u\in\mathcal{U}} C_{g,u}\leq U_g; \quad \sum\limits_{u\in\mathcal{U}} C_{d,u}\leq U_d, & \forall g\in\mathcal{G},\ \forall d\in\mathcal{D}; \\
\textit{Backhaul network constraints:} \\
\frac{1}{D_g}W_{\mathcal{B}}\log_2\left(1+\Gamma^{\mathcal{B}}_{g,d}\right)\geq R^{\mathcal{B}}_{\min}\cdot B_{g,d}, & \forall g\in\mathcal{G},\forall d\in\mathcal{D}; \\
\sum\limits_{g\in\mathcal{G}} B_{g,d}\leq 1, & \forall d\in\mathcal{D}; \\
\sum\limits_{d\in\mathcal{D}} B_{g,d}\leq D_g, & \forall g\in\mathcal{G}; \\
\textit{Access–backhaul constraints:} \\
C_{d,u}\leq\sum\limits_{g\in\mathcal{G}} B_{g,d}, & \forall d\in\mathcal{D},\forall u\in\mathcal{U}; \\
\sum\limits_{d\in\mathcal{D}}\sum\limits_{u\in\mathcal{U}} B_{g,d}\cdot C_{d,u}\leq U_g; & \forall g\in\mathcal{G}; \\
\textit{Drone air-space constraint:} \\
\Pi^d\in\mathcal{S}_d, & \forall d\in\mathcal{D}.
\end{cases}
\tag{5.8}
$$

Access network constraints: The first constraint guarantees that the access link rate between BS b and user u is above $R^{\mathcal{A}}_{\min}$ as soon as u accesses the network via b; the second constraint tells that a user cannot be covered by more than 1 BS; the third constraint accounts for the number of users that each BS (either *gNB* or *aBS*) can serve.

Backhaul network constraints: The forth constraint guarantees that the backhaul link rate between *gNB* g and *aBS* d is above $R^{\mathcal{B}}_{\min}$ as soon as d connects to the network via g; the fifth constraint tells that each *aBS* cannot connect to more than 1 *gNB*; the sixth constraint limits the number of drones that each *gNB* g can serve to a maximum of D_g drones.

Access–backhaul constraints: The seventh constraint states that a user u can connect to a drone d only if d is under the coverage of some *gNB*. Hence, each drone that provides network access to at least one ground user is connected to the network via one backhaul link, so that, every ground user served by a drone is indeed attached to the cellular network. The eighth constraint states that the number of users covered by those drones attached to the same *gNB* g is limited by the maximum capacity of users U_g in g.

Drone air-space constraint: Finally, the air location of a drone d has to be within a 3–D region $\mathcal{S}_d\subseteq\mathcal{S}\times[h_{\min},h_{\max}]$ which can be reached in the time interval used for

optimization, depending on flight speed and current drone position.

Feasibility. The optimization problem in Eq. (5.8) is always feasible. For instance, consider a general instance of the problem. Take a random position for each *aBS* d inside their reachable regions $\mathcal{S}_d$. Now set all binary variables $\{C_{b,u}\}$, $\{B_{g,d}\}$ equal to 0. Since the SINR functions $\Gamma^{\mathcal{B}}_{g,d}$, $\Gamma^{\mathcal{A}}_{b,u}$ are always positive (see Eqs. (5.6), (5.3), respectively), the solution satisfies all the constraints. This solution provides a utility function of 0 users covered, but it is feasible.

Complexity. Problem $\mathcal{C}$ is NP-Complete because, as shown in Appendix C, the well-known NP-Complete *Minimum-Geometric Disk-Cover (MGDC)* problem [125] can be reduced, in polynomial time, to a particular case of Problem $\mathcal{C}$.

The first constraint on the user access rate, when $b \in \mathcal{D}$, is non-linear and very complex (also the forth constraint), since it depends on the air-to-ground path-loss shown in Eq. (5.2) for one link, but also for the interfering links from other drones. To make the constraint more visual and remark its non-linearity and complexity, we develop its expression for a drone d as follows:

$$\frac{\frac{K_1}{(h^d)^2+r^2_{d,u}}\cdot 10^{K_2 P_{LoS}(h^d, r_{d,u})}}{N_u + \sum\limits_{d'\in\mathcal{D}\setminus\{d\}} \frac{K_1}{(h^{d'})^2+r^2_{d',u}}\cdot 10^{K_2 P_{LoS}(h^{d'}, r_{d',u})}} \geq (2^{K_3}-1)\cdot C_{d,u}, \tag{5.9}$$

where the continuous variable $r_{d,u} = \|(x^u, y^u) - (X^d, Y^d)\|$ is the distance between user u and the ground projection of drone d, and $K_1 = P^d_{Tx}\cdot\left(\frac{c_l}{4\pi f_c}\right)^2 \cdot 10^{\frac{\xi_{NLoS}}{10}}$, $K_2 = \frac{\xi_{NLoS}-\xi_{LoS}}{10}$ and $K_3 = \frac{U_d \cdot R^{\mathcal{A}}_{\min}}{W_{\mathcal{A}}}$ are constant. In Eq. (5.9) we see that this constraint depends on the position decision variables (X^d, Y^d, h^d) not only as an attenuation from the distance, but they also affect the LoS probability, as shown in Eq. (5.1).

Unlike previous works like [19, 119, 126], in which the drone-service condition is based *only* on the attenuation or the Signal-to-Noise Ratio (SNR), in (5.8) we have formulated a novel 3–D drone placement optimization that accounts for the actual inter-drone interference suffered by ground users.

Eq (5.8) represents a Mixed-Integer Non-Convex Program (MINCP), which is not tractable with currently available optimizers dealing with problems that are, at least, convex. Since problem (5.8) presents a non-convex formulation mainly because of the attenuation depending on the LoS probability, we cannot apply any off-the-shelf optimizer to optimally solve this problem. In addition, the problem itself is NP-Complete, so we resort to a heuristic, as detailed in Section 5.3.

5.2.2. Assignment of Fleet Destinations

The second problem to solve when users move and drones have to be repositioned is an assignment problem. Since it does not matter which *aBS* goes to which destination (as

long as such destination is reachable), we enforce each drone to fly towards the positions that minimize the aggregated flight-time by the fleet. Formally, we dispose of a fleet of D drones that must fly from source positions $\{\pi^d\}_{d\in\mathcal{D}}$ and reach target positions $\{\Pi^{d'}\}_{d'=1}^{D}$. Thus, we formulate the following *assignment problem*:

$$\begin{cases} \min\limits_{F_{d,d'}} U_{\text{fly}} = \sum\limits_{d\in\mathcal{D}}\sum\limits_{d'=1}^{D} \mathcal{T}\left(\pi^d, \Pi^{d'}\right)\cdot F_{d,d'}; & \\ \text{subject to:} & \\ \sum\limits_{d'=1}^{D} F_{d,d'} = 1, & \forall d\in\mathcal{D}; \\ \sum\limits_{d\in\mathcal{D}} F_{d,d'} = 1, & \forall 1\leq d'\leq D; \\ F_{d,d'}\in\{0,1\}, & \forall d\in\mathcal{D}; \\ & \forall 1\leq d'\leq D, \end{cases} \tag{5.10}$$

where we have introduced the binary variable $F_{d,d'}\in\{0,1\}$ to denote whether drone d flies from π^d to $\Pi^{d'}$ or not. $\mathcal{T}\left(\pi^d,\Pi^{d'}\right)$ is the assignment weight, and depends on the time that drone d needs to fly from source position π^d to the destination $\Pi^{d'}$. The equality constraints ensure that each destination $\Pi^{d'}$ is reached by only one drone d, and that each drone d reaches only one destination $\Pi^{d'}$. The utility U_{fly} is used to minimize the flight time of the fleet of drones.

For simplicity, we assume that drones d fly at a constant speeds of v_d. Thus, the time needed to fly from π^d to $\Pi^{d'}$ is:

$$\mathcal{T}^*\left(\pi^d,\Pi^{d'}\right) = \left\|\pi^d-\Pi^{d'}\right\| \Big/ v_d. \tag{5.11}$$

However, each drone d can reach only those destinations with coordinates within their reachable region $\mathcal{S}_d$. Hence, we impose infinite required time for each drone d to reach all destinations $\Pi^{d'}$ that fall outside $\mathcal{S}_d$:

$$\mathcal{T}\left(\pi^d,\Pi^{d'}\right) = \begin{cases} \mathcal{T}^*\left(\pi^d,\Pi^{d'}\right), & \text{if } \Pi^{d'}\in\mathcal{S}_d; \\ +\infty, & \text{if } \Pi^{d'}\notin\mathcal{S}_d. \end{cases} \tag{5.12}$$

Hence, the optimization in Eq. (5.10) assigns to each drone d one destination $\Pi^{d'}$ such that d is able to reach $\Pi^{d'}$ and the aggregated flight time is minimized.

Feasibility. The optimization problem presented in Eq. (5.10) is always feasible and finite. For instance, assigning to each drone d the destination $\Pi^d=(X^d,Y^d,h^d)$ obtained as an output of the optimization problem of Eq. (5.8) provides a (finite) feasible solution.

Complexity. The optimization problem in Eq. (5.10) is a special case of Mixed-Integer Linear Program (MILP), one that can be solved efficiently in polynomial time through the *Hungarian method* [127], with complexity $\mathcal{O}(D^3)$. Therefore, this

second problem related to dynamic networks is easy to address optimally in low-degree polynomial time.

5.3. Dynamic Drone Repositioning Algorithms

So far we have discussed the optimization of the placement of a fleet of *aBS*s hovering over a ground cellular network (see Section 5.2.1), and the optimal flight assignment that minimizes the flight time (see Section 5.2.2), thus overlaying a legacy cellular network managed by an orchestrator. Nevertheless, since users move, the optimization is reconsidered periodically, with updated drone air-space constraints. Repositioning has to be run frequently, so that we need efficient heuristics. Next, we describe how to practically implement the optimization framework described so far.

5.3.1. `OnDrone`: an Algorithm suit for On-demand Drone Coverage Optimization

Coverage Problem $\mathcal{C}$ is NP-Complete, thus optima cannot be reliably solved on-demand for fast placement of drones, which is key for dynamic repositioning cases. Thus, even if the problem was optimally solvable, the need of having an efficient heuristic would remain. To this aim, we propose here an On-demand Drone Coverage (`OnDrone`) algorithm, based on an *Extremal-Optimization* Algorithm (EOA) that runs in polynomial time. For benchmarking purposes, we further consider state-of-the-art proposals from [119] and [74].

5.3.1.1. `OnDrone` for multi-drone coverage

EOAs are evolutionary algorithms that restrict the search space and achieve near-optimal results in polynomial time [118]. EOAs are based on a fitness metric and, at each step, improve the configuration of the element of the system that yields the least contribution to the fitness metric. Therefore, EOA's principles perfectly match the nature of the coverage problem addressed. Specifically, the fitness function is the number of covered users and the least significant contribution comes from the drone that covers the least number of users. Such drone may be either far from users, where its transmissions are severely affected by the interference coming from the rest of *aBS*s, or in a position where the backhaul service is low. Thus, it is convenient to reposition such drone and increase the coverage.

The search space for drone locations is restricted to a lattice, as shown in Figure 5.3. We derive a cylindrical lattice that contains the ground network surface $\mathcal{S}$ composed by the positions over which `OnDrone` moves the *aBS*s to get the best coverage utility. Those lattice points that fall outside the ground region $\mathcal{S}$ are discarded, so the design of the

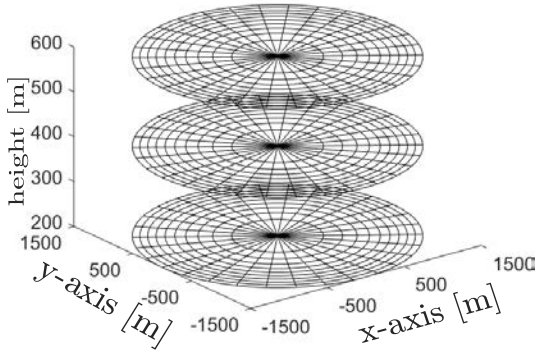


Figure 5.3: Cylindrical lattice with $N_\rho = 10$, $M_\theta = 30$, $H = 3$.

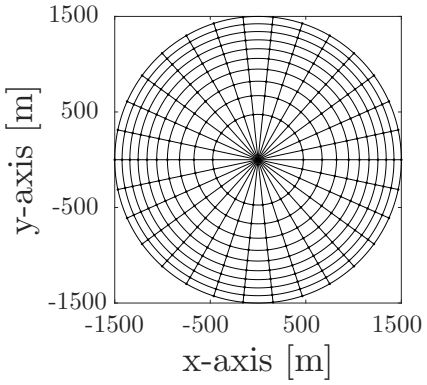


Figure 5.4: Circular base grid with $N_\rho = 10$, $M_\theta = 30$.

lattice applies to any shape of $\mathcal{S}$. We split the base into a grid of equal areas (see Figure 5.4), and the height into equidistant altitudes. This leads to a cylindrical lattice of equal volume subspaces. Specifically, we divide the square of the radius and the angle of the base grid in N_ρ and M_θ equal pieces. In polar coordinates, the resulting points and base grid are:

$$\begin{aligned} \rho_i &= \sqrt{i/N_\rho} \cdot R_C, \quad i = 1, ..., N_\rho; \\ \theta_j &= 2\pi \cdot (j-1)/M_\theta, \quad j = 1, ..., M_\theta; \\ \mathbb{G} &= \{(\rho_i, \theta_j) \in D(0, R_C) \mid 1 \leq i \leq N_\rho, 1 \leq j \leq M_\theta\}, \end{aligned}$$

where $D(0, R_C)$ is the closed disk in $\mathbb{R}^2$ centered at the origin and of radius R_C (where R_C is big enough to make $D(0, R_C)$ contain $\mathcal{S}$). In this way, the base area is divided into $N_\rho \times M_\theta$ regions with the same area $A = A_i = \frac{\pi}{M_\theta}(\rho_{i+1}^2 - \rho_i^2) = \frac{\pi R_C^2}{N_\rho M_\theta}$, for $i = 1, ..., N_\rho - 1$, which does not depend on i (see Figure 5.4). The height of the cylinder is divided into H equidistant segments in the interval between minimum and maximum drone hovering height, $h_{\min}$ and $h_{\max}$. In cylindrical coordinates, the resulting lattice is

$$\mathcal{L} = \{(\rho, \theta, h_k) \in \mathbb{G} \times \mathbb{R} \mid 1 \leq k \leq H\}. \tag{5.13}$$

The proposed EOA—`OnDrone`—begins with an initial feasible and suboptimal (random) implementation of the system. Then, it updates the positions of the *aBS*s providing worst individual contribution to the full performance, i.e., the drone covering less users. At each iteration, the "least fit" *aBS* is selected and *moved* to a reachable position where the system utility increases as much as possible, considering the coverage by ground- and drone-cells. Also, `OnDrone` provides a new position where the drone can be attached to a *gNB* that provides backhaul service with the guaranteed QoS. To decrease

Algorithm 5 `OnDrone`: On-demand Drone Coverage for 3-D Drone Placement

Input: Lattice $\mathcal{L}$, BSs $\mathcal{G}\cup\mathcal{D}$, users $\mathcal{U}$, Signal parameters.

1: Randomly place all d at $\Pi^d \in \mathcal{L}$. Define $\Pi = \{\Pi^d\}_{d\in\mathcal{D}}$.
2: Compute number $U_d^{'}(\Pi)$ of UEs covered by all $g\in\mathcal{G}$ and $d\in\mathcal{D}$ (If $\frac{1}{D_g}W_{\mathcal{B}}\log_2(1+\Gamma^{\mathcal{B}}_{g,d}) < R^{\mathcal{B}}_{\min}, \forall g\in\mathcal{G}$, then $U_d^{'}(\Pi)=0$).
3: Select $d_0 = \arg\min_{d\in\mathcal{D}}\{U_d^{'}(\Pi)\}$.
4: Take $\Pi^{d_0} = \arg\max_{\pi\in\mathcal{L}}\{\sharp$ of UEs covered if d_0 is at $\pi \mid \exists g\in\mathcal{G} : \frac{1}{D_g}W_{\mathcal{B}}\log_2(1+\Gamma^{\mathcal{B}}_{g,d_0}) \geq R^{\mathcal{B}}_{\min}$ in $\pi\}$.
5: If the same coverage remains when placing d_0 at Π^{d_0}, go back to step 3 ignoring unsuccessful d_0's.
6: Place d_0 at Π^{d_0} and set $\Pi \leftarrow \{\Pi^d\}_{d\in\mathcal{D}}$.
7: Go back to step 2 until:
- Ground-plus-air coverage is no longer improved.
- Maximum number of iterations i_0 is reached.

the probability of finding only local optima, we consider that if the worst performing drone cannot be moved to improve coverage, then we try to reposition the next worst-performing drone. `OnDrone` keeps moving the *aBS*s with lowest contribution until it does not find any better location for any *aBS*, or it reaches a maximum number of iterations $i_0\in\mathbb{N}$. The optimality of this algorithm is studied in Section 5.4.

Complexity. Algorithm 5 reports the pseudocode of the proposed `OnDrone`, in order to target maximum ground-plus-air users coverage, thus approximating the optimal solution of Coverage Problem $\mathcal{C}$. The complexity of `OnDrone` can be evaluated as follows. At each iteration, one drone $d_0 \in \mathcal{D}$ is selected and repositioned. Such drone d_0 selects the 3–D position Π^{d_0} in the lattice $\mathcal{L}$ (see Eq. (5.13)) at which *gNB*s and *aBS*s cover more users as long as there exists $g\in\mathcal{G}$ providing the guaranteed backhaul QoS in that position. A user u is covered by an *aBS* d only if the user rate experienced is greater than minimal user access rate $R^{\mathcal{A}}_{\min}$ and d is covering at most U_d users, so that u enjoys the guaranteed QoS. Thus, the signal strength from d_0 and from the rest of the drones in $\mathcal{D}\backslash\{d_0\}$ must be checked. This means that the complexity of each iteration is $\mathcal{O}(|\mathcal{L}|\cdot D\cdot U)$, where $|\mathcal{L}|$ is the size of the lattice. Since i_0 is the maximum number of iterations needed for the algorithm to converge to a solution, the complexity of `OnDrone` is $\mathcal{O}(i_0\cdot|\mathcal{L}|\cdot D\cdot U)$, where i_0 is constant and can be omitted.

We remark that, unlike the NP-Complete problem presented in the previous section, `OnDrone` requires a few iterations. As a matter of fact, `OnDrone` is intended to be used in an on-demand fashion, dynamically repositioning drones to adapt to user moves over time. `OnDrone` is then practical and can be executed at the network orchestrator.

5.3.1.2. `Seq`: Sequential Multi-Placement

In addition to `OnDrone`, we have also developed a simple heuristic that finds feasible solutions to the Coverage Problem $\mathcal{C}$ in polynomial time. We base this algorithm on the

`Efficient 3-D Placement`—hence the name of the algorithm—scheme derived in [119], and thus we adapt it to `Sequential Multi-Placement` (`Seq`) in order to support aerial networks with more than one *aBS*. In [119], the authors model the presence of one drone providing coverage in one single cell, i.e., they maximize coverage for single drone missions. They model a circular drone footprint, which allows to formulate a convex optimization problem due to the presence of *only* one drone. We adapt the proposal in [119] to place *aBS*s one by one, according to realistic interference metrics *not originally considered* in that work. `Seq` will be used as a benchmark for `OnDrone`.

We build the `Seq` heuristic by induction as follows: Since the framework proposed in [119] only considers drone-coverage, we first compute the amount of users covered by ground-cells. Second, we select one *aBS* and maximize coverage as described in [119] (i.e., using `Efficient 3-D Placement`) for the remaining non-covered users, namely $\mathcal{U}^{'}$. Here, the placement space for drones is restricted to those positions where at least one *gNB* provides backhaul connectivity with the guaranteed QoS. We denote as $\mathcal{U}_1$ to the set of users that are covered by the first selected *aBS*. For this first *aBS*, there are no interference issues.

Let $i > 1$ and assume that we have located $i-1$ *aBS*s and that we want to locate the i-th *aBS*. Assume that $\mathcal{U}_{i_k}$ are the sets of UEs that each previous i_k-th *aBS* covers at the moment of its placement. Then, `Seq` finds a 3–D position at which the i-th *aBS* covers more users from the set $\mathcal{U}^{'} \backslash \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$ and at least one *gNB* provides the requested backhaul QoS. Thus, the i-th *aBS* aims to cover the maximum number of users that are not covered yet.

`Seq` ends when the D-th *aBS* is placed. After this, the algorithm computes the actual number of users served according to interference (in both the backhaul and access network). Hence, `Seq` has the same objective as `OnDrone`: covering the maximum number of users according to QoS guarantees. We report the pseudocode in Algorithm 6.

At each iteration $i > 1$, the i-th *aBS* sequentially selects the best position for it based

Algorithm 6 `Seq`: `Sequential Multi-Placement`

Input: BSs $\mathcal{G} \cup \mathcal{D}$, users $\mathcal{U}$, and Signal parameters.

1: Compute the number of users covered by ground-cells.
2: Find $\Pi^{d_1} \in \mathbb{R}^3$ where d_1 covers more users from $\mathcal{U}^{'}$ and $\exists g \in \mathcal{G} \mid \frac{1}{D_g} W_{\mathcal{B}} \log_2(1 + \Gamma^{\mathcal{B}}_{g,d_1}) \geq R^{\mathcal{B}}_{\min}$.
3: Define the set of users covered by d_1: $\mathcal{U}_1 \subseteq \mathcal{U}^{'}$.
4: **for** $2 \leq i \leq D$
5: Find $\Pi^{d_i} \in \mathbb{R}^3$ where d_i covers more UEs in $\mathcal{U}^{'} \setminus \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$ such that $\exists g \in \mathcal{G} \mid \frac{1}{D_g} W_{\mathcal{B}} \log_2(1 + \Gamma^{\mathcal{B}}_{g,d_i}) \geq R^{\mathcal{B}}_{\min}$ in Π^{d_i}.
6: Define the set of UEs served by d_i: $\mathcal{U}_i \subseteq \mathcal{U}^{'} \setminus \bigcup_{1 \leq i_k < i} \mathcal{U}_{i_k}$.
7: **end for**
8: Derive the actual covered UEs according to interference.

on `Efficient 3-D Placement` from [119] maximizing its own coverage, no matter how its position affects the coverage of the remaining *aBS*s or the already set backhaul links. Since placing the new i-th *aBS* adds interference to the system, the previous $i-1$ drone-cells shrink, and cover less UEs than the ones originally intended by the `Seq` choice.

Complexity. The complexity of `Seq` is evaluated as follows. `Seq` makes D steps, one per each drone. At each step d, `Seq` defines the final position of drone $d \in \mathcal{D}$. In [119], the authors do not propose any algorithm for placing the drone, but they solve a convex mixed-integer non-linear program with a convex optimizer. Such optimizer does not run an algorithm with polynomial-time complexity. Instead, it performs a combination of interior-point methods with a Branch&Bound search [87]. Hence, we opt for approximating their problem through a search on the lattice $\mathcal{L}$. As in `OnDrone`, checking whether a user is covered requires to check the signal strength of the serving drone along with the interfering drones, i.e., D signal strengths. The complexity of this process is $\mathcal{O}(|\mathcal{L}| \cdot D \cdot U)$. After the last iteration, the algorithm checks which users are actually covered, since those users that at some iteration i were covered, may no longer be under coverage because of the repositioning of other drones in successive iterations. This check has a complexity of $\mathcal{O}(D^2 \cdot U)$. Thus, the complexity of the `Seq` algorithm is $\mathcal{O}(|\mathcal{L}| \cdot D \cdot U + D^2 \cdot U)$, that is similar to the complexity of `OnDrone` because $|\mathcal{L}| \geq D$, since drones cannot be co-located and so the number of possible distinct drone positions cannot be smaller than the number of drones (indeed, in a well designed system, $|\mathcal{L}| \gg D$).

5.3.1.3. `RA`: a `Repulsion-Attraction` scheme

Here we briefly describe the `Repulsion-Attraction (RA)` scheme derived in [74]. `RA` is a multi-drone placement scheme in which several self-organized *aBS*s dynamically change their position to track clusters of users. The approach is based on the assumption that *aBS*s will be attracted by the presence of users in the ground, and will be repulsed by *gNB*s and other *aBS*s in order to avoid interference.

Complexity. `RA` consists into maximizing a non-integer metric without any constraints. Hence, one can apply standard methods like line-search or trust-regions methods for unconstrained optimization. Such methods have low complexity and their performance depends on the target tolerance on the error. Moreover they converge really quickly [87], with a linear dependance on the number of iterations $i_{\texttt{RA}}$. Usually, $i_{\texttt{RA}}$ is inversely proportional to a convergence tolerance. The complexity is $\mathcal{O}(i_{\texttt{RA}})$.

5.3.2. *Bézier* Flight Routes

The last problem to solve consists in designing drone trajectories. The output of `OnDrone` (Section 5.3.1), and the *Hungarian method* (Section 5.2.2), provide the source and destination for each drone carrying an *aBS*. Therefore, we now design a route

optimization scheme.

While drones fly, both backhaul and user association change. Initially, each *aBS* attaches to a *gNB* with the required QoS and starts the flight. Once the backhaul QoS level is low due to long distance or interference impairment, the *aBS* sets a backhaul link with a new *gNB* with the guaranteed QoS. Similarly, while a drone flies, UEs attach to that *aBS* in case that the minimum access data link rate (i.e., QoS) is guaranteed. Upon arrival to the destination, the association is already optimal in terms of coverage.

On the one hand, drones have high aerial mobility and fly over a ground cellular network in a 3–D space, without many restrictions of walls, streets or vehicles. On the other hand, drones hovering over regions with good QoS from some *gNB*s or under-populated regions with a big surface may lead to under-utilized *aBS*s and low coverage, depending on the topology of users. Furthermore, if a drone is not fast enough, it might occur that when the drone arrives at the destination the user topology has changed too much so the destination is no longer optimal, and the network needs to be re-optimized. To avoid such undesired effects, and knowing that drones may have to be redirected while flying towards a destination, we propose drone paths following *Bézier curves* [79], instead of commonly assumed straight lines, as adopted in [78]. Indeed, using *Bézier curves* allows to deflect drone trajectories towards areas with higher user density, so to enhance drone coverage and enable unique coverage opportunities *while* drones seek their optimal position. Since we leverage the notion of *Bézier curve*, Appendix D provides some background on the subject.

In our proposal, we define the set of anchor points for our *Bézier*-based flight path and use the standard *de Casteljau* algorithm [79] to derive the *Bézier curve* corresponding to the selected anchor points (see Appendix D for details).

We obtain the set of anchor points inductively, as detailed next. Let π^d and Π^d be source and destination of a drone $d \in \mathcal{D}$, let $\omega > 0$ be the width for the two-sided offset region[4] of the curves and let $B > 1$ be the maximum number of anchor points for the *Bézier curve*. B is determined as the density of users covered per drone-cell, since in case a drone cannot cover more than B users, it does not make sense that such drone wishes to deflect its path attracted by more than B users. We take $\omega = 2R_d$, where R_d is the maximum range at which drone d can provide coverage in its optimal position. We define as the initial set of anchor points $\mathcal{P}_0$ both nodes: $\mathcal{P}_0 = \{\pi^d, \Pi^d\}$. Thus, we define the *Bézier curve* $\beta^{\mathcal{P}_0}(t)$ for $\mathcal{P}_0$, which is the segment joining π^d and Π^d, computed with the *de Casteljau* algorithm [79]. Now, we iteratively modify the current *Bézier curve* until we derive the final *Bézier* path. Given a curve $\beta(t)$, we denote its length as $\lambda(\beta(t))$, and take $\lambda\left(\beta^{\mathcal{P}_0}(t)\right)$ as a reference length for the final *Bézier curve*. Indeed, we build a *Bézier Scheme* such that the obtained curve is not longer than $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$, for a given $\alpha > 0$. α is determined as the fraction of the time interval in which a drone would

[4]The offset region is the area between the curve and its parallel, i.e., its offset curve.

Algorithm 7 *Bézier Scheme*

Input: $\mathcal{P}_0 = \{\pi^d, \Pi^d\}$, ω, B, $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$.

1: $\mathcal{U}^{\mathcal{P}_0} = \mathcal{U} \cap \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_0}(t)\right)$.
2: $k = 0$.
3: **while** $|\mathcal{P}_k| < B \quad \& \quad \lambda\left(\beta^{\mathcal{P}_k}(t)\right) < \tau$, **do:**
4: **for** $u \in \mathcal{U}^{\mathcal{P}_k}$, $g_u^{\mathcal{P}_k} = |\{u' \in \mathcal{U}^{\mathcal{P}_k} \mid \|u' - u\| \leq \omega/2\}| / (\pi\omega^2/4)$.
5: $u_{k+1} = \arg\max_{u \in \mathcal{U}^{\mathcal{P}_k}} \left\{ g_u^{\mathcal{P}_k} \mid \lambda\left(\beta^{\mathcal{P}_k \cup \{u\}}(t)\right) \leq \tau \right\}$.
6: $\mathcal{P}_{k+1} = \mathcal{P}_k \cup \{u_{k+1}\}, \quad \mathcal{U}^{\mathcal{P}_{k+1}} = \mathcal{U}^{\mathcal{P}_k} \cap \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_{k+1}}(t)\right)$.
7: $k = k + 1$.
8: **end while**
9: $\mathcal{P} = \mathcal{P}_{k-1}$.

have already arrived to its destination in case of following a straight path. Hence, we make sure that a drone reaches its destination but has the flexibility to deflect its path to improve coverage. Let $\mathcal{S}^{\omega}(\beta(t))$ be the two-sided offset region of width ω that results from stroking a curve $\beta(t)$, and let $\mathcal{U}^{\mathcal{P}_0} = \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_0}(t)\right) \cap \mathcal{U}$ be the set of users that are inside the offset region of $\beta^{\mathcal{P}_0}(t)$. We denote by $g_u^{\mathcal{P}_0}$ the gravity of a user $u \in \mathcal{U}^{\mathcal{P}_0}$ and define it as the density of users in a disk centered in u with radius $\omega/2$:

$$g_u^{\mathcal{P}_0} = \left|\{u' \in \mathcal{U}^{\mathcal{P}_0} \mid \|u - u'\| \leq \omega/2\}\right| \Big/ \left(\pi\omega^2/4\right). \tag{5.14}$$

The first user u_1 selected as anchor point is the one with highest gravity in $\mathcal{U}^{\mathcal{P}_0}$ such that the resulting *Bézier curve* is not longer than $\tau = (1+\alpha) \cdot \lambda\left(\beta^{\mathcal{P}_0}(t)\right)$, i.e.:

$$u_1 = \arg\max_{u \in \mathcal{U}^{\mathcal{P}_0}} \left\{ g_u^{\mathcal{P}_0} \mid \lambda\left(\beta^{\mathcal{P}_0 \cup \{u\}}(t)\right) \leq \tau \right\}. \tag{5.15}$$

Now, $\mathcal{P}_1 = \mathcal{P}_0 \cup \{u_1\}$ defines a new *Bézier curve*, $\beta^{\mathcal{P}_1}(t)$. The sorting of positions in the sets $\mathcal{P}_k$ is important, since each order defines a different curve. Thus, we sort the points in increasing order according to the distance to the source, π^d. Finally, $\mathcal{U}^{\mathcal{P}_1} = \mathcal{U}^{\mathcal{P}_0} \cap \mathcal{S}^{\omega}\left(\beta^{\mathcal{P}_1}(t)\right)$ is the updated set of users for next iteration. Then, we keep selecting anchor points for the *Bézier curve* while they exist, until the length of the *Bézier curve* exceeds τ or the maximum number B of anchor points is reached. At each iteration k, we build a new curve from $\mathcal{P}_{k-1}$, $\mathcal{U}^{\mathcal{P}_{k-1}}$ and $\beta^{\mathcal{P}_{k-1}}(t)$.

Complexity. We show the derived *Bézier Scheme* in Algorithm 7. Its complexity can be evaluated as follows. There is an iteration of the *de Casteljau* algorithm to derive each *Bézier curve* $\beta^{\mathcal{P}_k}(t)$. The complexity of the *de Casteljau* algorithm is quadratic with the number of anchor points of the *Bézier curve* that it builds [79]. The *Bézier Scheme* runs at most B iterations, and at each iteration k it uses $k+2 = |\mathcal{P}_k| \leq B$ anchor points. Thus, the complexity of the *Bézier Scheme* is $\mathcal{O}(B^3)$, where B is the maximum number of anchor points allowed.

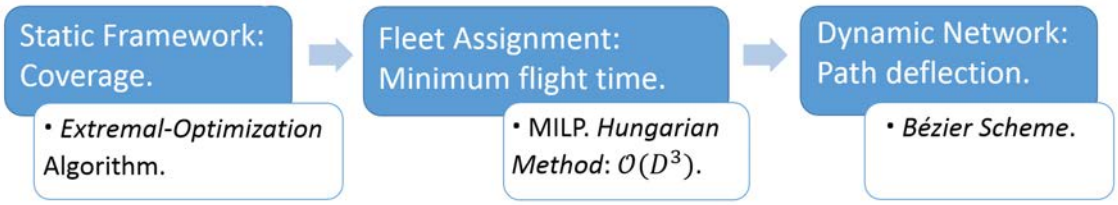


Figure 5.5: Flow chart of the drone-aided dynamic network.

5.3.3. Overall Complexity

In Figure 5.5 we show the flow chart of the proposed optimization framework for drone-aided dynamic cellular networks. As we have fully detailed, first we solve, through `OnDrone`, the framework for maximizing network coverage in polynomial time. `OnDrone` outputs the optimal 3–D positions at which drones must locate, given their previous positions. However, drones do not move instantaneously. The *Hungarian method* solves then the assignment problem that minimizes the distance flown in polynomial time, and outputs the source and destination of each drone concretely. Finally, our *Bézier Scheme* designs the flight routes of drones in order to deflect their paths towards clusters of users and increase the coverage efficiency over the time interval. This is another polynomial time algorithm. Therefore, the overall optimization framework runs in polynomial time. The overall complexity of our framework for fast repositioning of drone-aided cells is the result of summing the complexities of the schemes that compose such framework, which is shown in Figure 5.5. When the 3–D placement optimization is performed by `OnDrone`, the complexity is $\mathcal{O}(|\mathcal{L}|\cdot D\cdot U)+\mathcal{O}(D^3)+\mathcal{O}(B^3) = \mathcal{O}(|\mathcal{L}|\cdot D\cdot U)+\mathcal{O}(B^3)$, since $|\mathcal{L}| \gg U \gg D$.

In Table 5.3 we summarize the complexities of the detailed algorithms, including the *overall optimization framework*.

5.3.4. Orchestration of the Optimization Framework

The possibility to integrate the above described framework into a communication system raises several alternatives regarding where the different algorithms can be executed. First, there is the issue of deciding where the drones must be located at each time instant. This task (i.e., to execute `OnDrone` for drone placement) should be assigned to a device on which all drones have direct communication and therefore can easily know what must be their destination. Therefore, the *gNB*s seem to be an appropriate place for the orchestration of drones' positions. In alternative, and considering current trends in 5G networks architecture design, the orchestrator can be a software slice in the MEC, which is the edge-cloud computing platform of Fifth Generation (5G) cellular networks and which resides just next to base stations [128].

The second issue consists in designing drone trajectories, provided it must change its

Table 5.3: Summary of algorithms' complexity

Algorithm	***Complexity***
Optimal Aerial Coverage (Section 5.2.1)	NP-Complete
Assignment of Fleet Destinations (Section 5.2.2)	$\mathcal{O}(D^3)$
`OnDrone` (Section 5.3.1.1)	$\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U)$
`Seq` (Section 5.3.1.2)	$\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U + D^2 \cdot U)$
`RA` (Section 5.3.1.3)	$\mathcal{O}(i_{\texttt{RA}})$
Bézier Scheme (Section 5.3.2)	$\mathcal{O}(B^3)$
Overall Optimization Framework with `OnDrone` (Section 5.3.3)	$\boldsymbol{\mathcal{O}(\lvert\mathcal{L}\rvert \cdot D \cdot U) + \mathcal{O}(D^3)}$

current location. However, the *Bézier curves* used for the flight routes have been already designed in a discretized manner with the *de Casteljau* algorithm, using short straight segments to build a *Bézier curve*. At this point, we note that currently, most drones are already capable of autonomously travel to concrete positions following straight lines (see for instance [129]). So, drones can follow such short segments (using their originally integrated traveling mechanisms) without significantly deviating from the flight route provided by the *Bézier scheme*. In that case, those responsible for such tasks (i.e., obtain the discretized *Bézier curves* and follow the corresponding straight segments) are the drones themselves.

5.4. Experimental Results

Here we numerically assess the performance of our multi-drone optimization framework. We assess the coverage offered by the network, with the assistance of a fleet of drones, in a circular surface. We compare optimal placement results yielded by `OnDrone` (presented in Section 5.3.1.1) with the ones obtained with `Seq` (based on [119] and described in Section 5.3.1.2) and with the `RA` scheme (from [74], and described in Section 5.3.1.3). We also compare the results with the optimal achieved by means of Monte Carlo simulations, since computing exact optima is not doable in networks with as few as a fistful of UEs, *gNB*s, and drones. Besides, we compare our scheme to a modified `OnDrone` that neglects interference (referred as "`iNeg`" in the figures). Hence, we assess the importance of introducing interference in the analysis. We also compare coverage results while drones reposition following the *Bézier Scheme* (presented in Section 5.3.2) with a simpler *Straight Scheme*, as adopted in [78], which consists in moving drones over straight paths towards a certain destination identified with the *Hungarian method* (see Section 5.2.2). Since `RA` has been designed in [74] to dynamically track UEs, we compare

Table 5.4: Environmental parameters for the computation of LoS probability

Environment	***suburban***	***urban***	***dense***	***high-rise***
ξ_{LoS} [dB]	0.1	1	1.6	2.3
ξ_{NLoS} [dB]	21	20	23	34
a	4.88	9.61	12.08	27.23
b	0.43	0.16	0.11	0.08

also the *Bézier* and *Straight Schemes* when the path planning is based on `RA`.

We mainly study the placement and repositioning of drones in synthetic and realistic scenarios, and the effects of interference and LoS on user coverage over time, under four classes of environmental scenarios: *suburban*, *urban*, *dense* and *high-rise*. These four environments correspond to different densities of elements (e.g., buildings) that affect the LoS probability. Moreover we study three distinct cases of deployment scenarios, in which the location of users follows different distributions:

- *Poisson Point Process (PPP)*: We place UEs in a circular surface, according to a Poisson point process.

- *Cheese*: We define a surface that includes certain regions which are not accessible to UEs, and locate UEs uniformly random in the rest of the network. Then, we have a surface with empty areas (like in a Swiss cheese). This user distribution is typical of public gardens or areas with restricted zones.

- *Capital*: We also run our numerical evaluation over a simplified map of the center of a dense capital city, Madrid, considering main zones of people affluence and no users in indoor installations.

In the *PPP* and *Cheese* scenarios, we locate *gNB*s according to the same distribution. Heterogeneous synthetic distributions would be of interest for traffic demand-based optimizations [130], which is out of the scope of this work, and we therefore do not consider them. However, in the *Capital* scenario, we consider the actual locations of those *gNB*s that belong to the main network operator in the city.

Table 5.4 gathers the parameters for the air-to-ground path-loss model and the probability of LoS described in Eqs. (5.1) and (5.2), depending on the density of the environment that we consider in the numerical simulations. These parameters are obtained based on the number of buildings and large signal obstructions per unit area, building height distribution, ratio of built-up area and clean surfaces, etc., as it has been derived in [67], based on the ITU recommendations [122]. Such parameters allow to differentiate the main four environmental conditions.

Table 5.5 gathers the parameters that, unless otherwise specified, we have used for the network model, regardless of the simulation environment. We take a circular surface of $R_C = 1.5$ km of radius where there are 10 *gNB*s, and a height range between $h_{\min} = 60$ m

Table 5.5: System and simulation parameters

Parameter	***Value***
Circular surface radius R_C	1500 m
Height range, $[h_{min}, h_{max}]$	$[60, 600]$ m
gNB, *aBS* Tx power, P^d_{Tx}, P^g_{Tx}	10 dBm, 44 dBm
Users guaranteed QoS	0.72 Mbps
*gNB*s Carrier frequency, $f_{\mathcal{B}}$	1815.1 MHz
*aBS*s Carrier frequency, $f_{\mathcal{A}}$	2.63 GHz
gNB, *aBS* Bandwidths	20 MHz
Thermal Noise Power	-174 dBm/Hz
HPBW of *gNB*s	65 degrees
Time interval length T	60 secs
UEs, Drones speed	2 m/s, 15 m/s
Lattice dimensions N_ρ, M_θ, H	20, 40, 40
Monte Carlo runs per instance	10^7
Instances of simulations	1000

and $h_{\max} = 600$ m for drones. This is a generally doable height range,[5] since lower values would be too close to ground (and, e.g., vehicles or even people) and higher elevation would be affected by high-speed winds which are unsafe for an aerial network of simple drones. However, in our numerical evaluation the actual maximum drone altitude is rather determined by the environment density. For instance, in a *high-rise* environment, although high altitudes increase the probability of LoS, the attenuation is much stronger, so that drones need to fly closer to the ground. In contrast, the *suburban* or *urban* environments do not suffer strong attenuation, so that drones can fly higher. However, a high altitude turns into links with higher LoS probability, thus yielding more interference for far ground users.

The power transmission from *aBS*s is 10 dBm, as adopted in [131–133], which is notably lower than the usual 44 dBm used for *gNB*s in the ground network (as we adopt). This is because *aBS*s have much higher LoS probability than ground base stations and do not use omnidirectional antennas, and hence require much less power. Using higher *aBS* transmission power, as 25 or 44 dBm, may provide better coverage due to better signal strength, although also provides less resilience to interference impairment, as we discuss in our results. Moreover, *aBS*s are carried by flying drones, which spend most of their energy into hovering, flying towards desired positions at a given speed, and carrying the weight of the communication equipment. This poses serious constraints on transmission power, as evaluated in [131]. Hence, we have chosen to use a 10 dBm of power transmission for analyzing our framework and algorithms. The guaranteed user

[5]Such under-kilometer altitudes comply with current possibilities of commercial drones. For instance, DJI Phantom 4 has an elevation range of few thousands of meters, according to its commercial specifications.

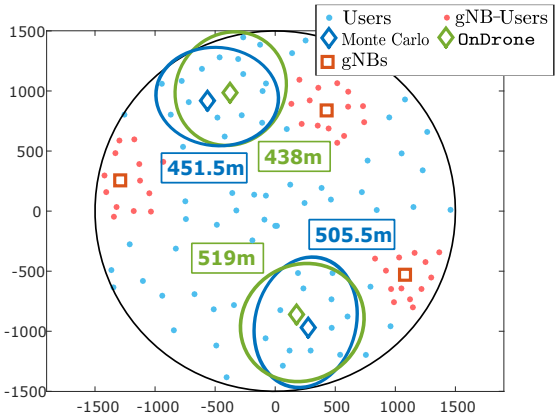


Figure 5.6: Drone 3–D placement. $D=2$, $U=100$. Scenario: *urban, PPP*.

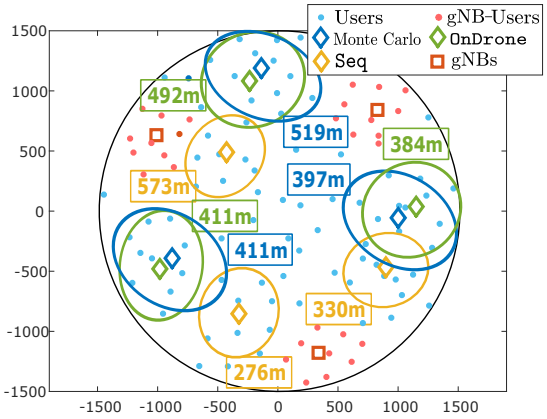


Figure 5.7: Drone 3–D placement. $D=3$, $U=100$. Scenario: *dense, PPP*.

data rate is 0.72 Mbps, which guarantees video streaming with 240p resolution [134], and allows 360p and 480p resolution in many cases over the MPEG-4 standard. It also allows adequate videoconferencing quality using video compression [135]. Such guaranteed rate, with customary 20 MHz bands and assuming that no more than 100 users can attach to a BS, corresponds to guaranteeing that the SINR is higher than $\gamma_{\mathcal{A}}=10.9$ dB, according to the Shannon capacity formula (see Appendix E.1 for a discussion on the minimum data rate experienced under coverage). Besides, such SINR value allows for a 16QAM modulation and a coding rate of 1/2 in LTE communications, as derived in [136], although we also study the impact of using other guaranteed rates. The carrier frequency used by *gNB*s (either for backhaul channels between *gNB*s and *aBS*s or access channels between *gNB*s and UEs) and for the access channel between *aBS*s and users are 1815.1 MHz and 2630 MHz, respectively. These channel parameters are as in the LTE/LTE-A network provided by Movistar in Madrid. As it has been discussed in [120], the antenna patterns of *gNB*s for backhaul connectivity are directional with a HPBW of 65 degrees, according to the 3GPP technical specifications [137]. For dynamic experiments, we slot time into intervals of length $T=60$ s. This means that every minute, we re-optimize the network by means of `OnDrone` (or with any of the benchmark schemes, as `Seq` or `RA` schemes) and immediately re-direct drone flights for dynamic repositioning using the solution of our assignment problem and a flight route computed with either the *Bézier* or the basic *Straight Scheme* from [78]. Users move according to the Random Way-Point (RWP) model[6] (see [138, 139]) with an average speed of 2 m/s. We update user positions every second, while drones fly at a constant speed of 15 m/s over a continuous path. Moreover, such flight speed allows for low drone energy consumption, according to the energy consumption model for aerial aircrafts derived in [131].

We use MATLAB `R2018a` to simulate channel conditions and mobility of drones and

[6]The RWP model is one of the most studied and used mobility models to assess mobile networks. It is simple and easy to implement.

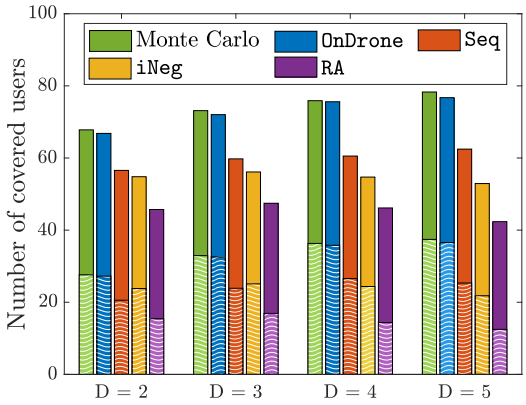


Figure 5.8: Comparison of algorithms on total coverage (solid bars) and *aBS* coverage (stripped bars), $U = 100$. Scenario: *dense, PPP*.

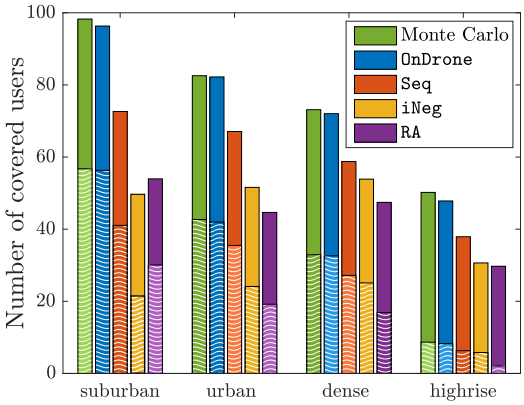


Figure 5.9: Impact of environment on total coverage (solid bars) and *aBS* coverage (stripped bars) coverage. $D = 3$, $U = 100$. Scenario: *PPP*.

UEs. For small networks, we have performed Monte Carlo simulations to search for the optimal drone placement. For each network instance, we have run 10^7 random positions for the fleet and taken the settings for the best coverage. Such simulations are very time-consuming even for small networks, yet we have observed that such number of Monte Carlo runs per instance provides a coverage output that is hard to improve, since the output placement provided remains invariable for a large number of additional runs, as we have preliminary tested. We have simulated every scenario 1000 times in order to derive the statistics shown in the figures.

As mentioned earlier, the high complexity of the exact solution of the Coverage Problem $\mathcal{C}$, allows us to find optima only for small instances of the problem, with a reduced number of *gNB*s, *aBS*s and UEs. Instead, `OnDrone` only requires a few iterations to converge. Thus, for realistically larger deployments, we only show the results obtained with `OnDrone`, and compare the results to what achieved by `Seq` based on [119], the `RA` scheme [74], and a modified `OnDrone` scheme that neglects inter-*aBS* interference ("`iNeg`" in the figures). We have evaluated the coverage performance with denser lattices in `OnDrone` and observed that the results cannot be significantly improved (since they are already close to optimal, as shown in the comparison with the optimal placements), while imposing more computational complexity. We also evaluate the impact of *Bézier* routes vs. straight flying routes [78]. The flight assignment problem is solved optimally and efficiently by the *Hungarian method* [127], so we do not comment on its performance. At the end of the section we also provide a summary of the lessons learnt from our performance evaluation study.

5.4.1. Coverage Optimization

Here we numerically evaluate the coverage performance at a precise time instant. We pictorially show the drone footprints and indicate the altitude, computed with the different schemes that target coverage maximization, namely with the optimum (i.e., with its approximation obtained by means of Monte Carlo simulations), with `OnDrone`, and with `Seq`. With `OnDrone`, we clearly see in Figure 5.6 that *aBS*s are positioned very close to the optimal positions, while in Figure 5.7 we see that the `Seq` scheme locates drones in much distinct positions (as well as the `RA` scheme, not shown here to keep the figure clear). Indeed, `Seq` makes a greedy decision at each step for a given drone. Hence, `Seq` finds a good position for such drone knowing the interference incurred by the previously located drones. However, the additional interference from drones located afterwards is ignored, which results on shrunk coverage areas as the final performance. Also, the altitudes provided by Monte Carlo simulations and `OnDrone` are only slightly different. In general, *aBS*s avoid locations already covered by *gNB*s.

Another important aspect to pay attention to is the shape of the drone footprints. Unlike in currently used models [19,64,66,140], the area served by a drone is not circular, due to inter-drone interference, which is not considered in the mentioned works. In Figure 5.8, we show the average number of covered users in a network with 100 users distributed in the ground surface according to a *PPP*, and different fleet sizes. Solid bars represent the total amount of covered users, while stripped bars represent users covered by *aBS*s. The figure shows that `OnDrone` approximates well the optimal solution (within a mere 1% from reaching the optimal) both to total coverage and *aBS* coverage, while neglecting interference leads to very inaccurate results, getting worse as the fleet size increases (and hence backhaul and inter-drone interference increases). In the figure, we also see that `Seq` only covers around 80% of the optimal coverage, depending on the fleet size. `RA` is not even able to outperform the coverage results from `Seq`, and its performance soon decays due to interference issues in the presence of as few as 5 *aBS*s. This shows that although it is practical, the intuition behind `RA` is not accurate enough for optimizing coverage.

In Figure 5.9 we further compare the coverage achieved when considering the four reference environments of Table 5.4, for the same *PPP* case discussed above. The LoS likelihood between *aBS*s and UEs decreases in denser environments. Thus, drone-cells shrink, and *aBS*s cover less users. Indeed, the figure shows a factor ∼6 between the *aBS* coverage achievable in *suburban* and *high-rise* environments, and also that the ground network handles higher total coverage percentages as the environment grows denser, at least for small fleet sizes. Here, we again see the high accuracy of `OnDrone` in comparison to the optimum searched by means of Monte Carlo simulations, in both total and *aBS* coverage. As before, `Seq` and `RA` provide coverage noticeably lower. We also see that neglecting interference is very counter-productive for *aBS* coverage in low

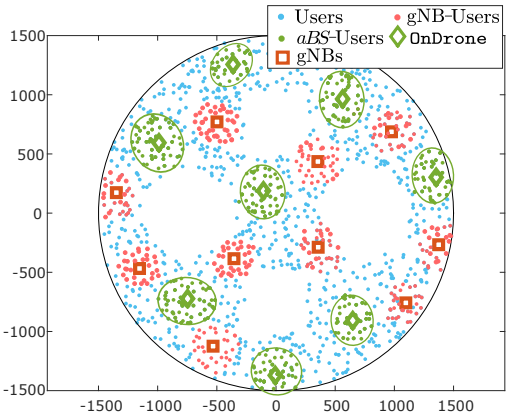


Figure 5.10: Drone 3–D placement. $D = 8$, $U = 1000$. Scenario: *dense, Cheese*.

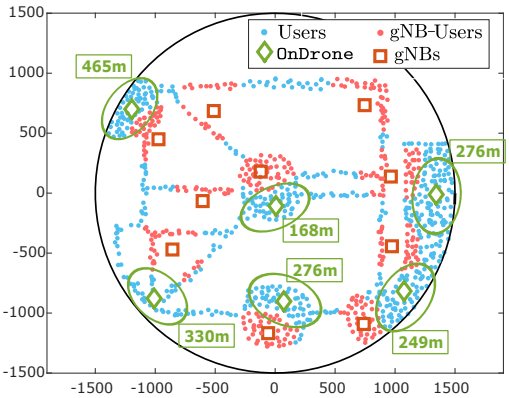
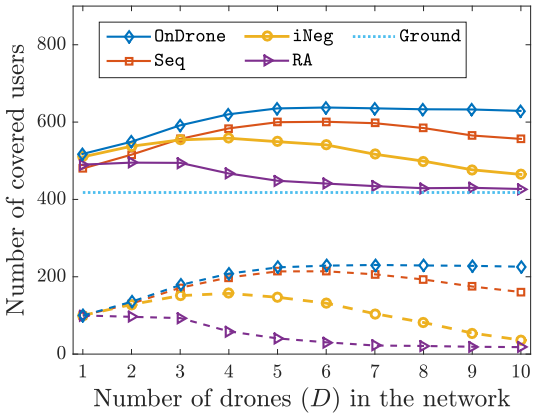


Figure 5.11: Drone 3–D placement. $D = 6$, $U = 1000$. Scenario: *dense, Capital*.

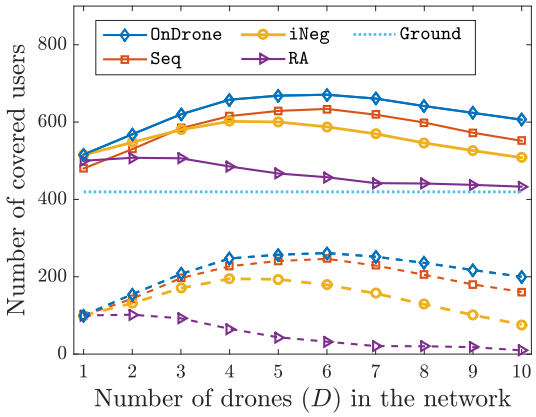
dense environments, since there is higher LoS probability and links are attenuated easily from interfering signals. The opposite behavior is observed with `RA`, which provides reasonable results for *suburban* scenarios (better than `Seq`) but then its performance decays with denser environments (and `RA` becomes the worst scheme). Results derived in the *Cheese* and *Capital* scenarios are qualitatively similar to those discussed above for the *PPP* scenario. Thus, we omit the results here.

To give a performance sample of `OnDrone` for larger fleets, Figures 5.10 and 5.11 show the placement of 8 and 6 *aBS*s in a *dense Cheese* and *Capital* scenario, respectively. Here, drones tend to follow UEs distribution, avoiding empty areas and regions covered by *gNB*s. Indeed, `OnDrone` avoids also overlapping drone-cells, thus incurring low inter-*aBS* interference and being able to cover more users.

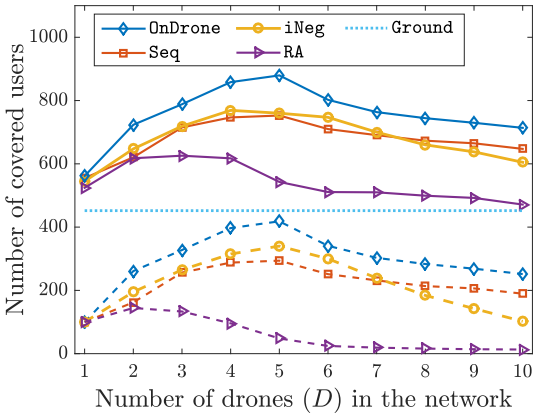
In Figure 5.12 we further study the impact of the fleet size on coverage, with $U = 1000$ users, for each scenario. Here, solid lines represent total coverage and dashed lines report the portion of users that would be served by *aBS*s. We also show coverage performance in absence of drones, labelled as `Ground` in the figures. First, for the *PPP* scenario in Figure 5.12(a), adding more drones increases total and *aBS* coverage because there are more *aBS*s that can cover larger areas with limited interference. Each analyzed scheme behaves significantly different. When the fleet size becomes larger than $D = 7$, the coverage remains stable or even slightly decays for `OnDrone`. Here, we notice that the larger the fleet is, the more interference issues appear in both the backhaul and the access network. `OnDrone` is able to maintain a stable coverage by reallocating positions to drones to have good backhaul connectivity while providing stable and wide coverage to users, thanks to the design based on *extremal-optimization*. However, `Seq` does not have a design that allows a reconfiguration of the aerial network. Hence, it suffers more from interference in the access and backhaul sides. The figure also shows that neglecting interference leads to very poor coverage, as well as with the `RA` scheme. In both cases, adding drones



(a) Scenario: *dense, PPP.*



(b) Scenario: *dense, Cheese.*

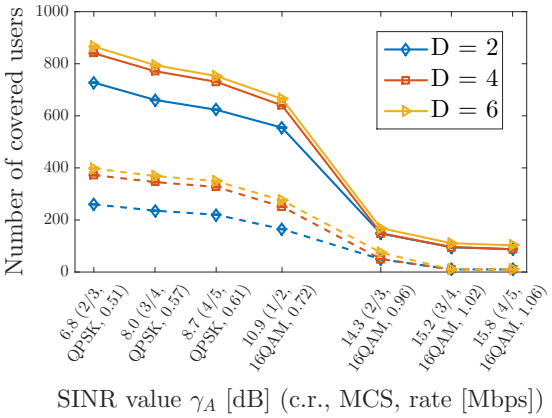


(c) Scenario: *dense, Capital.*

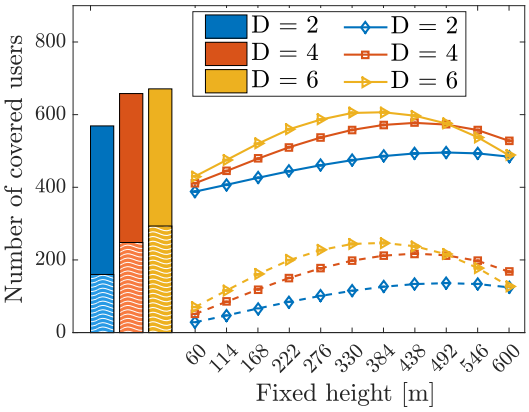
Figure 5.12: Total coverage (solid lines) and *aBS* coverage (dashed lines) for $U = 1000$ UEs.

becomes soon counter-productive. While we have adapted in `Seq` the scheme proposed in [119] to account for interference, the `RA` scheme proposed in [74] does not target any specific interference metric in the computation of its *repulsion* component. This explains why `RA` behaves poorly with as few as 4 or more *aBS*s in a *dense* scenario. Instead, as we have checked numerically, when the interference is neglected—or approximated by a constant value—the coverage apparently never stops increasing with the fleet size. In fact, without interference, having more drones implies covering more non-interfering drone-cells. However, in reality, it happens that *there is an exact number of drones that maximizes coverage, depending on the environment.*

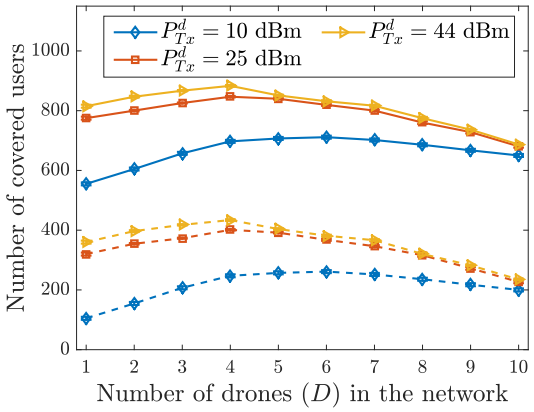
In Figures 5.12(b) and 5.12(c) we show the same type of graph for the *Cheese* and *Capital* deployment, respectively. The *Cheese* case confirms that `OnDrone` is a good option for irregular deployments, in which the area to cover in the ground surface is



(a) Impact of SINR value $\gamma_{\mathcal{A}}$ in total coverage (solid lines) and *aBS* coverage (dashed lines).



(b) Impact of fixed height in total coverage (solid lines) and *aBS* coverage (dashed lines). The bars show total (solid) and *aBS* (stripped) coverage with adaptive height.



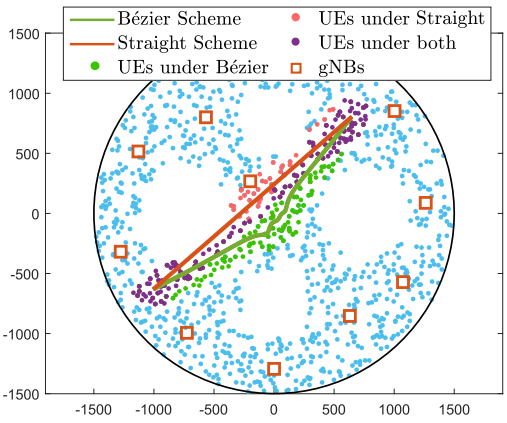
(c) Impact of drone transmission power P^d_{Tx} in total coverage (solid lines) and *aBS* coverage (dashed lines).

Figure 5.13: Study of tunable network parameters: Guaranteed bandwidth, fixed drone height h^d, and drone transmission power P^d_{Tx}. $U = 1000$ UEs. Scenario: *dense, Cheese.*

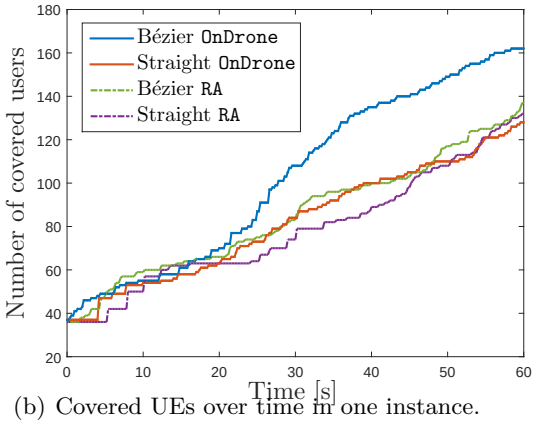
smaller than the complete circular surface, so drone positions are more packed. Here, we observe more clearly the effect that increasing the fleet size is not beneficial for coverage purposes, since the surface $\mathcal{S}$ is irregular, so that users are more packed and hence it is harder for *aBS*s to fully avoid interference. Indeed, `Seq` is not able to avoid interference as good as `OnDrone`, and suffers larger performance drops with more than 5 drones, as well as the scheme neglecting interference. `RA` appears to be able to manage interference issues, although with no good coverage results overall. In the *Capital* case, no matter the adopted scheme, once the optimum number of *aBS*s is reached, it is hard to add more drones without incurring interference, although in this scenario `Ondrone` substantially overcomes the rest of schemes. This is due to the fact that users are concentrated in relatively small areas and nearby drones can interfere large masses of users. In any case,

`OnDrone` largely outperforms any other benchmarking schemes also in realistic networks as the one extracted from a dense capital city (Madrid).

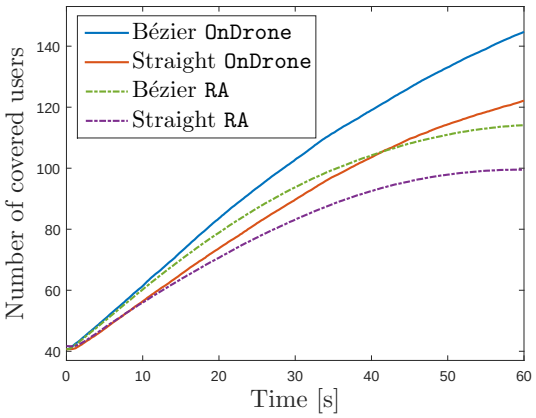
So far we analyzed our framework in comparison with significant state-of-the-art proposals, and shown that our approach provides significant gain. We now show, in Figures 5.13(a)–5.13(c), user coverage obtained with `OnDrone` when tuning a few key parameters. Specifically, in a *dense Cheese* deployment, we consider that case in which the guaranteed user data rate varies based on the SINR value $\gamma_{\mathcal{A}}$, the drone height becomes fixed, and the *aBS* power transmission increases, respectively. We show in all cases both total and *aBS* coverage. In Figure 5.13(a), we consider fleet sizes of $D = 2, 4, 6$ drones with different user data rate guarantees. Such SINR values correspond to the Modulation and Coding Scheme (MCS) values marked in the figure, according to [136]. Here, we observe that $\gamma_{\mathcal{A}} = 10.9$ dB is a good election: on the one side, lower $\gamma_{\mathcal{A}}$ values provide higher total and *aBS* coverage since the QoS requirement is less strict, but the MCS is only QPSK, which renders a considerably lower final user throughput; on the other side, the highest QoS requirements lead to much better MCS and coding rate ("c.r." in the figure), but here the QoS requirement gets too strict so that coverage performance falls notably. In Figure 5.13(b), we fix the drone height to altitudes between 60 and 600 meters, for fleet sizes of $D = 2, 4, 6$ drones. Moreover, in the left side of the figure we show a histogram with the total and *aBS* coverage results when the height is left as an adaptive choice of `OnDrone`, as intended by the framework proposed in this chapter. The results show that, depending on the fleet size, there is an optimal fixed height for the fleet where signal strength is good and the interference impairment remains stable, hence providing the best coverage. However, the difference with respect to the case in which the height is adaptive is around 20% for *aBS* coverage, which supports the idea that flexible non-uniform altitudes are convenient for drone-aided networks. In Figure 5.13(c), we analyze the impact of transmission power. We have selected three typical values for P_{Tx}^{d}: 10 dBm [131, 132], 25 dBm [76, 141], and 44 dBm [142] (besides, 44 dBm is the power transmission used for *gNB*s in cellular networks). Here, we clearly see that with 25 and 44 dBm the coverage is significantly increased due to better signal strength from the serving drone. However, fleet size increases, the framework is not able to keep the coverage stable and the interference quickly impairs coverage performance. Conversely, a 10 dBm power transmission allows the framework to combat the interference from multiple *aBS*s and keep the coverage stable. This shows that lower power transmissions make the framework more resilient. Moreover, since aerial networks are very energy-limited [131], using high power transmission needs to guarantee that the performance is more energy-efficient. Note also that the difference in Watts from 10 to 25 dBm is more than 96%, while the corresponding coverage improvement is only 30% in the best case.



(a) Example of drone trajectories.



(b) Covered UEs over time in one instance.



(c) Average covered UEs over time.

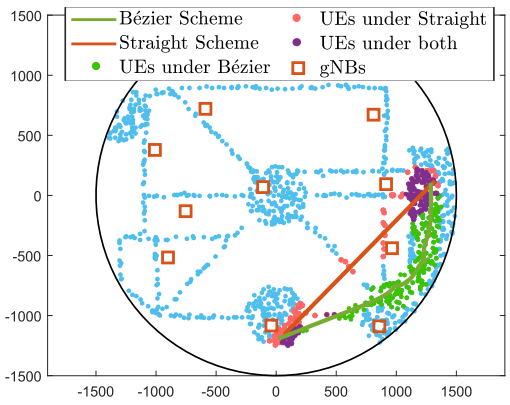
Figure 5.14: Continuous repositioning. $D = 1$, $U = 1000$. Scenario: *dense, Cheese*.

5.4.2. Continuous Repositioning

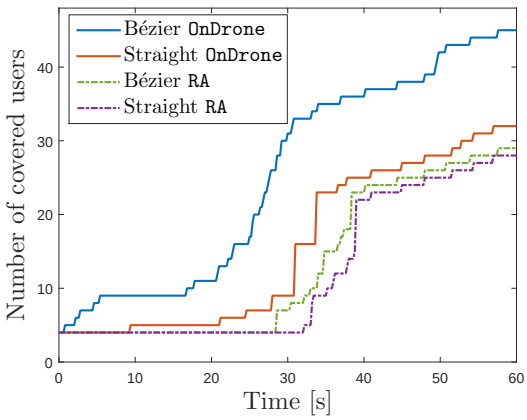
Having assessed the basic properties and performance figures of `OnDrone` for drone placement in Section 5.4.1, we now consider flight routes. Specifically, we study the performance of `OnDrone` and `RA` (specifically proposed for dynamic cases) with repositioning routes computed with either the *Bézier Scheme* or with the *Straight Scheme* every $T = 60$ s, in two practical and realistic scenarios: *Cheese* and *Capital*. Irregular topologies like *Cheese* and *Capital* are more interesting to study with respect to regular *PPP* cases because they clearly provide visual fact of the importance of our *Bézier Scheme* or alike schemes using deflected routes when several regions have really low densities of users.

In order to assess dynamic topologies, we consider a random initial position of drones in the network. In the successive time intervals, the source position of each drone is the last location it was occupying in the previous interval.

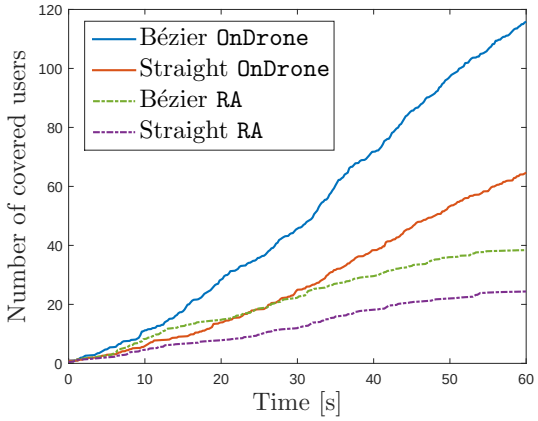
In order to compute the *Bézier curves* used as flight routes, we keep adding anchor



(a) Example of drone trajectories.



(b) Covered UEs over time in one instance.



(c) Average covered UEs over time.

Figure 5.15: Continuous repositioning. $D = 1$, $U = 2000$. Scenario: *high-rise*, *Capital*.

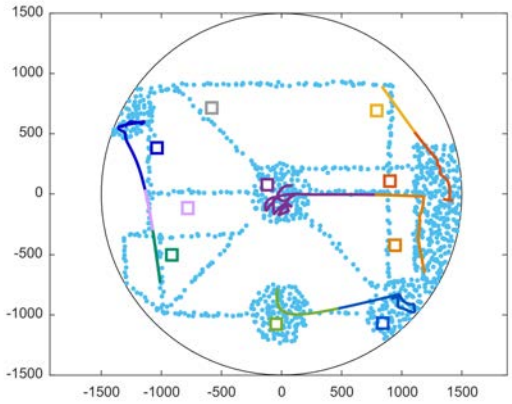
points and run iterations of the *de Casteljau* algorithm in the *Bézier Scheme* until the segments of the piece-wise curve approximating the *Bézier curve* are all shorter than 3 m, which corresponds to flight segments of 0.2 s. Hence we can consider the network as practically static in each of such segments, and solve the corresponding coverage problem.

In Figures 5.14 and 5.15 we analyze the performance of continuous *aBS* repositioning with the *Bézier Scheme*, in comparison to the *Straight Scheme* in the *dense Cheese* and *high-rise Capital* scenarios, respectively. In both cases, we show an illustration of drone trajectories (subfigure (a)) with the corresponding *aBS* coverage over time (subfigure (b)) and the average *aBS* coverage for the scenario obtained with longer and repeated simulation runs (subfigure (c)).

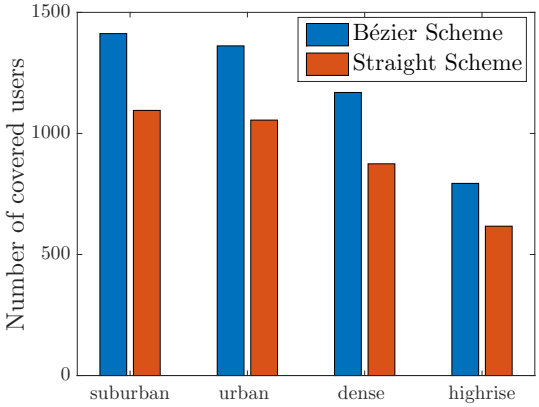
In Figure 5.14(a) we show an instance of a network surface where four empty regions have no user (or few users) demanding for coverage, and 1 drone provides coverage assistance. While a drone following straight lines can be barely used during the full time

interval since (i) it traverses part of the empty regions and (ii) it does not avoid ground cells covered by fixed *gNB*s, the *Bézier Scheme* deflects drone routes towards regions with denser population, avoiding empty regions and *gNB*-served users. The resulting gain of a simple example is quantified in Figure 5.14(b). The figure depicts the count of *aBS*-covered users over time for 60 s (the count is updated at each segment of the piece-wise curve approximating the *Bézier* path, every 0.2 s). In contrast with the *Straight Scheme*, in this example the *Bézier Scheme* provides a coverage gain from the beginning of the interval, and shows a gain of 26% after about 45 seconds. The gain remains high until the drone ends its path. The figure also shows that `OnDrone` largely outperforms `RA`. Indeed, *Bézier* and *Straight Schemes* used with `RA` most of the time perform worse than `OnDrone` with the *Bézier Scheme*. These coverage performance gaps vary depending on the instance. Hence, to fairly compare each scheme, in Figure 5.14(c) we report the *average* cumulated *aBS* coverage with each scheme, calculated over the entire numerical simulation of the scenario and over the tested initial positions. On average, by using *Bézier* flight paths in combination with `OnDrone`, at the end of an interval T, *aBS* coverage increases by a remarkable 18% compared to *Straight Scheme*. Using *Bézier* paths is beneficial also when using the `RA` placement algorithm. However, since the coverage guarantees of `RA` are much weaker than with `OnDrone`, the achievable coverage is lower.

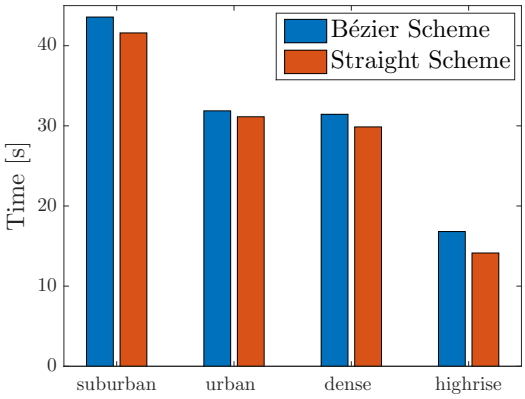
In Figure 5.15(a) we show a snapshot of drone trajectories optimized with the *Bézier Scheme* over *high-rise* environmental conditions in the *Capital* scenario with 1000 UEs located uniformly at random over the city, plus five masses of 200 UEs. Again, to make the presentation visual and simpler, we use a case with a single drone, although the results for more drones are similar. In the figure, we also report the corresponding flight path computed with the *Straight Scheme*. We observe that the *aBS* flies from the bottom towards the right side of the city, where there are more users. Moreover, the *Bézier* path avoids two *gNB*s in order not to overlap *aBS* coverage with *gNB* coverage. In addition, although not shown to keep the figure clear, the *Bézier* path meets two masses of 200 users on its way to the final destination, where another mass of 200 users is targeted also by the *Straight Scheme*. However, the path followed under the *Straight Scheme* does not avoid coverage overlapping with *gNB*s and includes regions with very low density of users, hence missing key coverage opportunities. Figure 5.15(b) quantifies the gain due to deflected drone paths for an example over these conditions. The figure shows that the *Bézier Scheme* considerably increases the cumulated number of users covered since the beginning of the interval, reaching a coverage efficiency increase of 33% with respect to the *Straight Scheme*, and with minimal route adjustments. As done in the *dense Cheese* scenario, to fairly compare the performance obtained with either *Bézier* or straight paths, in Figure 5.15(c) we report the *average* gain in terms of the count of users covered since the beginning of a time interval T. We see that, on average, by using *Bézier* flight paths in combination with the `OnDrone` placement algorithm, at the end of an interval T the



(a) *Bézier* trajectories. Each solid line represents one drone. Trajectory segments are colored with the color of the backhaul associated *gNB*. Scenario: *high-rise*.



(b) Average number of covered users per minute. *Bézier Scheme* vs. *Straight Scheme*.



(c) Average time spent per user per minute in coverage. *Bézier Scheme* vs. *Straight Scheme*.

Figure 5.16: Continuous repositioning during 10 minutes, $D = 4$, $U = 2000$. Scenario: *Capital*.

aBS coverage is increased by a remarkable 47%. Using *Bézier* paths is beneficial also when using the `RA` placement algorithm. However, since the coverage guarantees of `RA` are much weaker than with `OnDrone`, the achievable coverage is much lower.

To show the repositioning operation of multiple drones with `OnDrone` and the *Bézier Scheme*, in Figure 5.16(a) we show the trajectories of 4 drones in the *high-rise Capital* scenario, during an interval of duration $10\,T$. Here, we capture the main behaviours of our repositioning schemes. The figure reflects backhaul *gNB* associations by showing multi-color paths, each color corresponding to a different *gNB* indicated on the map with a marker of the same color. Drones at the bottom and left side start at a suboptimal position. On their path, they deflect over more populated zones and follow streets that connect the origin and final destinations. Once they get to optimal positions, they remain hovering the zone, adjusting according to changes in the presence of users. The drone

starting at the top behaves similarly, although it does not benefit from following any street from source to destination because (*i*) this street is partially covered by the yellow *gNB* and (*ii*) user density is low. The remaining drone has the longest path and traverses several regions. First, it flies upwards to comply with the optimal position of the first stages, but soon deflects its path towards the center of the city where a new optimal position is identified, and remains hovering this central region over the rest of the time. We also see that when this drone is very close to the red *gNB*, it does not switch its backhaul association because the yellow *gNB* provides enough connectivity while the drone coming from the top has no alternative but to attach to the red *gNB*. Pictorially showing correlation between drone trajectories and user movements is not simple with the scenarios presented. However, for a simple case with a dozen of users and one *aBS*, we refer the reader to the illustrative example described in Appendix E.2.

Finally, in Figures 5.16(b) and 5.16(c) we analyze, for the same experiment with four repositioning drones, the *aBS* coverage count (i.e., the average number of users covered by drones in a one-minute time window) and the average time during which an *aBS*-covered user receives service within the same one-minute time window. We use the *Capital* scenario in all possible environmental density cases, with 2000 users in total. Figure 5.16(b) shows that the *Bézier Scheme* provides a coverage gain around 25% with respect to using straight paths. More interestingly, the denser environments present higher coverage gains. While the *urban* environment provides a gain of 25%, the *dense* environment increases such gain up to a remarkable 33%. Therefore, deflected paths are more important in denser scenarios, like in historical districts of old cities and in modern downtowns.

The number of seconds spent under *aBS* coverage, reported in Figure 5.16(c), tells the quality of coverage opportunities offered by drones *on the move*. The figure shows little coverage time differences between the *Bézier* and *Straight Schemes*, with some improvements observed with the former. Therefore, we can conclude that the increased coverage count offered by using deflected flight paths is not obtained at the expenses of the time spent by users under coverage. In general, we remark that although the *Bézier Scheme* needs more time to reach the optimal placement identified with `OnDrone`, it outperforms significantly approaches based on straight flights for *aBS*s.

5.5. Lessons Learnt and Discussion

Our analysis has shown that optimizing the position of a fleet of drones in a coordinated manner is unfeasible with standard solvers. However, using *extremal-optimization* techniques, we have seen that it is possible to achieve nearly-optimal results in polynomial time. Thus, it is doable to run drone repositioning on a minute time scale. Indeed, our results show that with realistic topologies, drones are able to follow mobile

masses of people over time, or either react quickly to changes, thanks to `OnDrone`.

The performance of `OnDrone` that we have proposed and validated depends on the thickness of the lattice used to reduce the space of options used to seek (near-)optimal drone positions. Denser lattices would result in improved accuracy, although at the expense of computational cost. With as much as 10 drones and a few thousands of users, it is possible to achieve accurate near-optimal results using commodity hardware, as we have done for our numerical evaluation. For larger fleets, more powerful hardware is required, which is not a big problem for base stations and definitely not a problem for cellular back-offices, e.g., in the MEC of 5G networks [143].

Our numerical results have quantitatively shown the importance of integrating a fleet of drones in a cellular network. The presence of inter-drone interference is key in the optimization of drone positions, so that it cannot be neglected, contrary to what so far used in the literature. The presence of interference makes the optimal number of drones finite, with no advantages coming from deploying unnecessarily dense fleets.

We have also seen that repositioning is a key component of the overall drone-aided network framework. Repositioning requires solving not only for 3–D drone positions, but also finding a flight assignment and deflect flight routes. We have used the *Hungarian method* to solve the assignment problem optimally and *Bézier curves* to obtain very efficient and dynamic trajectories that offer coverage opportunities to many users without reducing their time under drone coverage. Such dynamic behavior is key for network surfaces in which some regions cannot host users that can benefit of the presence of *aBS*s, e.g., in forests or in indoor installations in residential and commercial areas, and also to avoid flying over those ground regions already served by *gNB*s.

Indeed, our dynamic *Bézier Scheme* presents remarkable results, and we have observed that it enhances a lot coverage experienced in realistic topologies, especially in densely populated cities with *dense* or *high-rise* profiles. Where main avenues and landmarks attract users, our *Bézier Scheme* allows the drones to easily follow masses of users on selected paths and areas.

In general, we have shown that it is feasible to have an autonomous dynamic aerial network that reorganizes itself to optimize network coverage.

6 α-Fair Throughput Optimization with the Aid of Drone Relays

In this chapter, we address the optimization of 3-D positions for a fleet of coordinated drone relays aiding a set of ground BSs, as depicted in Figure 6.1, aiming at a fair throughput distribution among users. We base the optimization on the α-fairness metric [144], which is a high-level generalization of fairness metrics. The parameter α can be tuned to analytically target, e.g., maximum throughput, proportional fairness or max-min fairness with a single framework [145]. In the analysis, we model transmission technology details like the random variations of signal quality received by users over time, the interference caused by relays and BSs, the use of slotted time-frequency resources, the wireless backhaul attachment, and cell selection and resource allocation procedures. Specifically, we adopt stochastic models for path-loss and availability of LoS and NLoS links, and cast our problem into an OFDMA-like resource allocation scheme with several constraints.

The problem of finding the exact optimal drone positions is NP-complete. In particular, our analysis unveils that the role of interference caused by drones and the stochastic characterization of LoS between drones and ground users make the optimization problem intractable. However, we show that the problem can be addressed by leveraging EO algorithms, which are a class of algorithms specifically designed for polynomial time optimization with intertwined variables [118]. The EO operation is based on picking the "least fit" element of a discrete set and change its configuration parameters in order to improve a global utility function. We therefore formulate a suitable utility function, targeting α-fair user throughput across the network, and design a Parallelized Alpha-fair Drone Deployment (`PADD`). `PADD` iteratively updates the position of the least fit drone, i.e., the drone relay station that contributes the least to the utility function. We validate our algorithm and evaluate its performance by means of simulations of realistic static and dynamic scenarios. As an illustration of dynamic cases, we evaluate the performance of our algorithm when customers move towards a stadium before a sport event, so that their density grows over time. Beside illustrating the advantages offered by `PADD` over state

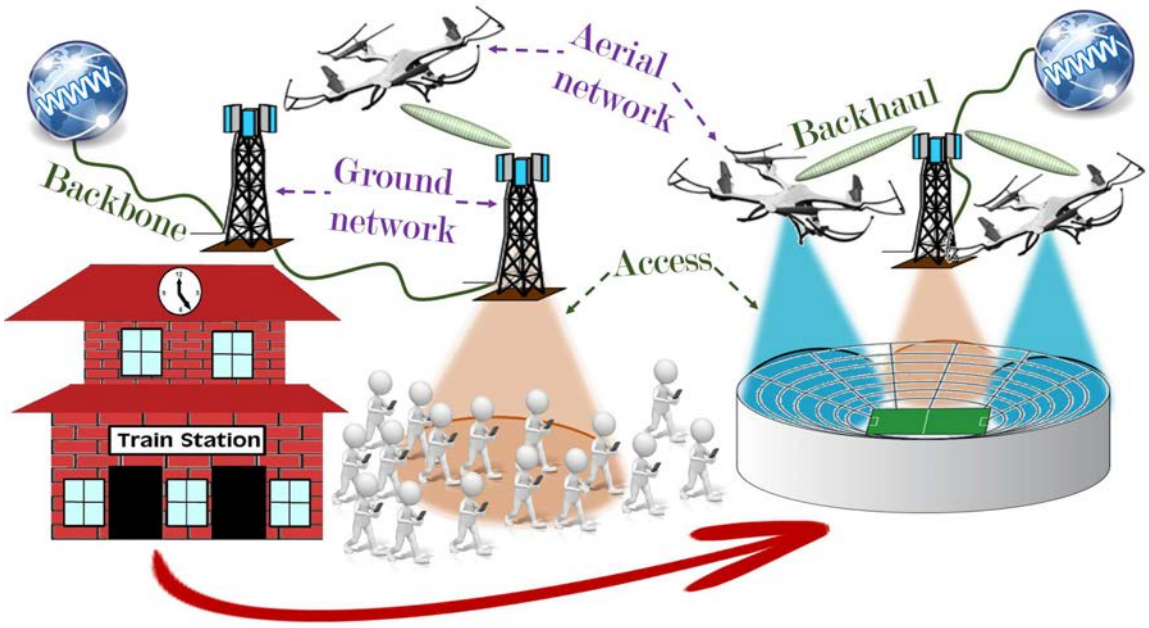


Figure 6.1: Reference scenario: multi-drone-aided network.

of the art algorithms, our numerical results show that optimizing network throughput without considering fairness is not beneficial in dense environments since drones serving the same area generate too much interference.

The main contributions of this chapter are summarized:

- We propose a dynamic drone relay-aided network in which we maximize the network capacity in terms of α-fairly distributed resources and throughput rates among ground users and backhauled aerial base stations.
- We show that the problem is NP-Complete.
- We propose `PADD`, an approximation algorithm based on *extremal-optimization* that solves the optimization problem in low-degree polynomial time.
- We propse `PADD`, an approximation algorithm based on *extremal-optimization* that solves the optimization problem in low-degree polynomial time.
- We derive closed-form solutions to optimal α-fair network throughput of static networks with several backhauled wireless relays per wired ground base station as a key component to be integrated into `PADD`.
- We asses our proposals over a real topology of a dense city and compare the provided dynamism with state-of-the-art solutions.

The rest of the chapter is structured as follows. Section 6.1 presents the system model and Section 6.2 derives the framework for optimizing drone positions under the α-fairness

metric. Section 6.3 describes the design of our optimization algorithm, while Section 6.4 provides numerical results. Section 6.5 discusses the findings of this chapter and possible practical implementation issues.

6.1. System Model

Our goal is to derive an analytical framework that finds optimal 3-D locations of drone relay stations, given the position of users and ground stations. We target α-fair instantaneous user throughput, hence, jointly with reference scenario and assumption on path-loss and interference in both access and backhaul, we also provide details on how users perform cell selection and get resources allocated.

To measure transmission capacity, we use the Shannon formula $W \log_2(1+\gamma)$. Thus, each link can have a different capacity, depending on bandwidth (W) and SINR experienced (γ).

6.1.1. Reference Scenario

The reference scenario considered in this Chapter follows the same network assumptions as in Chapter 5, yet we find a few differences, as the target here is different. As in Chapter 5, we consider a flat ground surface $\mathcal{S}$ where a set $\mathcal{G}$ of G ground base stations, referred to as *gNB*s, provide cellular service with known position $\Pi^g = (X^g, Y^g)$. We further assume that every *gNB* g is wired to the internet with a backbone capacity τ_g. A set $\mathcal{U}$ of U UEs is on the ground, requesting cellular service, with known position $\pi_u = (x_u, y_u)$. Also, the network disposes of a fleet $\mathcal{A}$ of A *aBS*s that act as mobile relays mounted on a drone. As drones fly in the air space, we denote as $\Pi_a = (X_a, Y_a, h_a)$ the 3-D position of drone a. We denote as $\mathcal{B} = \mathcal{G} \cup \mathcal{A}$ the set of all the base stations that form the whole network.

Once we have repointed these preliminary details for the system model, we remark that all details regarding adopted bandwidth spectrum and channel and interference modelling for air and ground connections are adopted as in Chapter 5 (see Section 5.1.2).

6.1.2. Cell Selection and Resource Allocation

BSs cannot provide service to unlimited numbers of users because (i) available radio resources are limited and (ii) it is necessary to guarantee a minimum set of radio resources to each connected user, to guarantee signaling exchange with the BS; this is needed to schedule data transmissions to and from the BS. Of course, the number of devices is also limited by the minimum bandwidth that the system aims to guarantee to each user. Therefore, in general, the maximum number of users that can be simultaneously served is limited, and we denote by $U_{\max}$ such number. We assume that users perform cell selection

as in LTE networks [146]: first, UEs select the BS with strongest SNR observed; if the request is rejected because channel conditions deteriorate or the BS runs at maximum capacity, then the UE performs cell re-selection, and tries to attach to the BS with next strongest SNR, and so on until the user gets attached. Note that the best-SNR policy adopted here is the one currently adopted in cellular networks and it is based on the availability of a Channel State Indicator (CSI) at the UE [146].

For what concerns resource allocation, we assume that *gNB*s and *aBS*s schedule cellular users according to an OFDMA system. Today's BSs use an OFDMA system and dispose of a finite set of physical resource blocks organized in subframes, which repeat to form frames lasting a few milliseconds (1 to 10 ms in 3GPP-compliant networks). A physical resource block is the smallest unit of time-frequency resources that can be allocated to a user. Thus, we assume that the minimum bandwidth allocated to a user is the bandwidth corresponding to one resource block and the scheduler guarantees that each user receives, on average, at least one block per subframe in each OFDMA frame. We denote as $W_{\mathcal{G}}^{\min}$ and $W_{\mathcal{A}}^{\min}$ the minimum bandwidth that a *gNB* or an *aBS* can allocate to a single user.

Backhaul links also use an OFDMA system, although *aBS*s select a *gNB* according a the global network optimization criterion rather than based on SNR. Moreover, each backhaul link (g, a) disposes of a minimum bandwidth $W_{\mathcal{B}}^{\min}$ to relay traffic.

6.2. Optimization

Here we derive an analytic framework for the 3-D positions of *aBS*s , to optimize throughputs based on α-fairness. Depending on the value of $\alpha \geq 0$, known as the α-fairness level, the metric captures different fairness criteria such as weighted proportional fairness ($\alpha = 1$), max-min fairness ($\alpha \to +\infty$) or the maximum capacity ($\alpha = 0$).

We formulate the drone positioning problem as a MINCP.

6.2.1. Utility with α-Fairness

The α-fairness metric is a mathematical function of a set of resources that are shared among several entities that depends on the parameter $\alpha \in \mathbb{R}$ [144]. Denoting as $T_{b,u}$ the throughput of user u attached to BS $b \in \mathcal{B} = \mathcal{G} \cup \mathcal{A}$, we define the α-fair throughput utility $\mathcal{U}_{\text{thr}}^{\alpha}$ as:

$$U_{\text{thr}}^{\alpha} = \begin{cases} \sum\limits_{u \in \mathcal{U}} \left(\sum\limits_{b \in \mathcal{B}} T_{b,u} \right)^{1-\alpha} \cdot \frac{1}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{u \in \mathcal{U}} \log \left(\sum\limits_{b \in \mathcal{B}} T_{b,u} \right), & \alpha = 1. \end{cases} \tag{6.1}$$

Since a user u only connects to one BS at a time, there can be only one non-zero valued $T_{b,u}$ for each user u. Hence the utility function is additive in terms of utilities

conveyed by single BSs. With the above, we seek not only optimal *aBS* positions and BS–UE associations, but also optimal backhaul association and optimal allocation of physical resources.

6.2.2. Problem Formulation

Here, we formally present the optimization problem addressed in this chapter.

Throughput Problem $\mathcal{T}$*: Given a set $\mathcal{G}$ of G fixed gNBs, a fleet $\mathcal{A}$ of A relay aBSs hovering at heights in the range $[h_{\min}, h_{\max}]$, a set $\mathcal{U}$ of U ground UEs that may connect to either a gNB or an aBSs, each of which can serve $U_{\max}$ UEs at most, find the optimal position of each aBS $a \in \mathcal{A}$, the optimal user association, the optimal backhaul association and the optimal user resource allocation so to maximize the α-fair throughput utility function.*

On the access network side, we denote as $C_{a,u} \in \{0,1\}$ the decision variable that tells whether u connects to *aBS* a. Similarly, $C_u^{g_u} \in \{0,1\}$ tells whether u connects to *gNB* g_u. Decision variables $W_{b,u}$ and $T_{b,u}$ denote bandwidth and throughput allocated to link (b,u). The throughput is a decision variable and not directly computed with the Shannon formula, because, in addition to bandwidth limitations we must account for access, backhaul and backbone bottlenecks.

On the backhaul network side, we denote as $B^{g,a} \in \{0,1\}$ the decision variable that tells whether *aBS* a is attached to *gNB* g. Variables $W^{g,a}$ and $T^{g,a}$ denote bandwidth and throughput of the backhaul link (g,a), respectively. The resulting optimization program is presented in Eq. (6.2).

$$
\begin{cases}
\max\limits_{\Pi_a, C_{a,u}, W_{b,u}, B^{g,a}, W^{g,a}} U_{thr}^{\alpha} = \begin{cases} \sum\limits_{u\in\mathcal{U}} \left(\sum\limits_{b\in\mathcal{B}} T_{b,u} \right)^{1-\alpha} \cdot \frac{1}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{u\in\mathcal{U}} \log\left(\sum\limits_{b\in\mathcal{B}} T_{b,u} \right), & \alpha = 1; \end{cases} \\
\text{s.t.:} \\
\textit{gNB–aBS association constraints:} \\
\sum\limits_{g\in\mathcal{G}} B^{g,a} = 1, \quad \sum\limits_{a\in\mathcal{A}} B^{g,a} \leq A_g, \qquad \forall g \in \mathcal{G}, \forall a \in \mathcal{A}; \\
\textit{gNB–aBS capacity constraints:} \\
W_{\mathcal{B}}^{\min} \cdot B^{g,a} \leq W^{g,a} \leq W_{\mathcal{B}} \cdot B^{g,a}, \qquad \forall g \in \mathcal{G}, \forall a \in \mathcal{A}; \\
\sum\limits_{a\in\mathcal{A}} W^{g,a} \leq W_{\mathcal{B}}, \qquad \forall g \in \mathcal{G}; \\
T^{g,a} \leq W^{g,a} \log_2\left(1 + \gamma_{g,a}^{\mathcal{B}}\right), \qquad \forall g \in \mathcal{G}, \forall a \in \mathcal{A}; \\
\textit{gNB backbone constraint:} \\
\sum\limits_{a\in\mathcal{A}} T^{g,a} + \sum\limits_{\substack{u\in\mathcal{U}: \\ g=g_u}} T_{g,u} \leq \tau_g, \qquad \forall g \in \mathcal{G}; \\
\textit{BS–UE association constraints:} \\
C_u^{g_u} + \sum\limits_{a\in\mathcal{A}} C_{a,u} = 1, \qquad \forall u \in \mathcal{U}; \\
\sum\limits_{\substack{u\in\mathcal{U}: \\ g=g_u}} C_u^g \leq U_{\max}, \quad \sum\limits_{u\in\mathcal{U}} C_{a,u} \leq U_{\max}, \qquad \forall g \in \mathcal{G}, \forall a \in \mathcal{A}; \\
\textit{gNB–UE capacity costraints:} \\
W_{\mathcal{G}}^{\min} \cdot C_u^{g_u} \leq W_{g_u,u} \leq W_{\mathcal{G}} \cdot C_u^{g_u}, \qquad \forall u \in \mathcal{U}; \\
\sum\limits_{u\in\mathcal{U}} W_{g,u} \leq W_{\mathcal{G}}, \qquad \forall g \in \mathcal{G}; \\
T_{g_u,u} \leq W_{g_u,u} \log_2\left(1 + \gamma_{g_u,u}^{\mathcal{G}}\right), \qquad \forall u \in \mathcal{U}; \\
\textit{aBS–UE capacity constraints:} \\
W_{\mathcal{A}}^{\min} \cdot C_{a,u} \leq W_{a,u} \leq W_{\mathcal{A}} \cdot C_{a,u}, \qquad \forall a \in \mathcal{A}, \forall u \in \mathcal{U}; \\
\sum\limits_{u\in\mathcal{U}} W_{a,u} \leq W_{\mathcal{A}}, \qquad \forall a \in \mathcal{A}; \\
T_{a,u} \leq W_{a,u} \log_2\left(1 + \gamma_{a,u}^{\mathcal{A}}\right), \qquad \forall a \in \mathcal{A}, \forall u \in \mathcal{U}; \\
\sum\limits_{u\in\mathcal{U}} T_{a,u} \leq \sum\limits_{g\in\mathcal{G}} T^{g,a}, \qquad \forall a \in \mathcal{A}; \\
\textit{Air space constraint:} \\
\Pi_a \in \mathcal{S}_a, \qquad \forall a \in \mathcal{A}.
\end{cases}
\tag{6.2}
$$

Constraints in Eq. (6.2) correspond to the following restrictions:

***gNB–aBS* association constraints** state that every *aBS* must associate with one *gNB* to set a wireless backhaul link, and that every *gNB* can serve at most A_g *aBS*s.

***gNB–aBS* capacity constraints** impose guarantees on bandwidth and throughput allocation on backhaul links.

The ***gNB* backbone constraint** restricts every *gNB* to provide connected *aBS*s and connected users with an aggregated throughput not higher than the capacity τ_g of the backbone link that serves that *gNB*.

BS–UE association constraints state that each user can associate only to one BS, either a ground station or a drone, and that the maximum allowed number of UEs served by each BS is limited to $U_{\max}$.

***gNB*–UE capacity constraints** impose guarantees on bandwidth and throughput allocation on *gNB*–UE links.

***aBS*–UE capacity constraints** impose guarantees on bandwidth and throughput allocation on *aBS*–UE links, while at the same time the throughput of a user cannot exceed the backhaul link capacity and the aggregate user throughput cannot exceed the backhaul throughput.

The **air space constraint** delimits the 3-D air space in within which an *aBS* can be moved, and which is a ball centered in the current position of the drone with a radius equal to the distance that the drone can fly within a fixed time (i.e., time itself is the real constraint).

Modeling air-to-ground connections brings unavoidable non-convex functions, so that the formulated problem is not tractable with currently available optimizers, which are able to deal *only* with problems that are convex.

Moreover, the problem of finding the exact optimal drone positions is NP-Complete. Indeed, the NP-Complete MGDC problem [125] can be reduced, in polynomial time, to a special instance of the problem where users get 1 bps if a drone serves them and 0 bps otherwise. This result is a direct consequence of the NP-Completeness proof of the Coverage Problem that we have presented in Chapter 5, because coverage can be seen as a particularly simplified throughput problem using an on/off, SINR-threshold-based, throughput function.

6.3. Extremal Optimization

The optimization framework proposed in Section 6.2 is non-convex and mixed-integer, hence not solvable with any off-the-shelf optimizer [147], not even with emerging methods like geometric programming, which does not work for mixed-integer programs [148]. The problem is hard to solve because any change in a decision variable (e.g., a position of a drone) affects backhaul and user association as well as resource allocation for all users in

the network. This is the kind of problems EO has been thought for.

To find time-efficient and near-optimal solutions, we propose a Parallelized Alpha-fair Drone Deployment (`PADD`) algorithm. We base the design of `PADD` on decoupling the problem in the four main decisions that the optimization framework must make: (*i*) the 3-D positions of the fleet of *aBS*s; (*ii*) the sets of users attached to each *gNB* and *aBS*; (*iii*) *gNB*–*aBS* backhaul association; (*iv*) bandwidth allocation to backhaul links *gNB*–*aBS* as well as to access links from each *gNB* or *aBS* to their attached users.

In what follows, we formally describe the `PADD` operation and provide details of each step that `PADD` takes. The algorithm iteratively solves four steps, as pictured in Figure 6.2 as a flow-chart: after deriving an initial feasible system setting, the least fit *aBS* a_0 is selected in order to locally probe non-searched positions that improve the current network performance. While probed positions do not provide a relative improvement of $\delta \geq 0$ over the current network performance, new local non-searched positions are probed. In case a probed position improves performance, *aBS* a_0 is *moved* to such position and the current system performance is updated. In case this new performance provides a relative improvement higher than $\varepsilon \geq \delta$, the new least fit *aBS* is selected and the process begins again. Otherwise, the algorithm converges and outputs the decided system setting. We next formalize the operation of `PADD` more in detail.

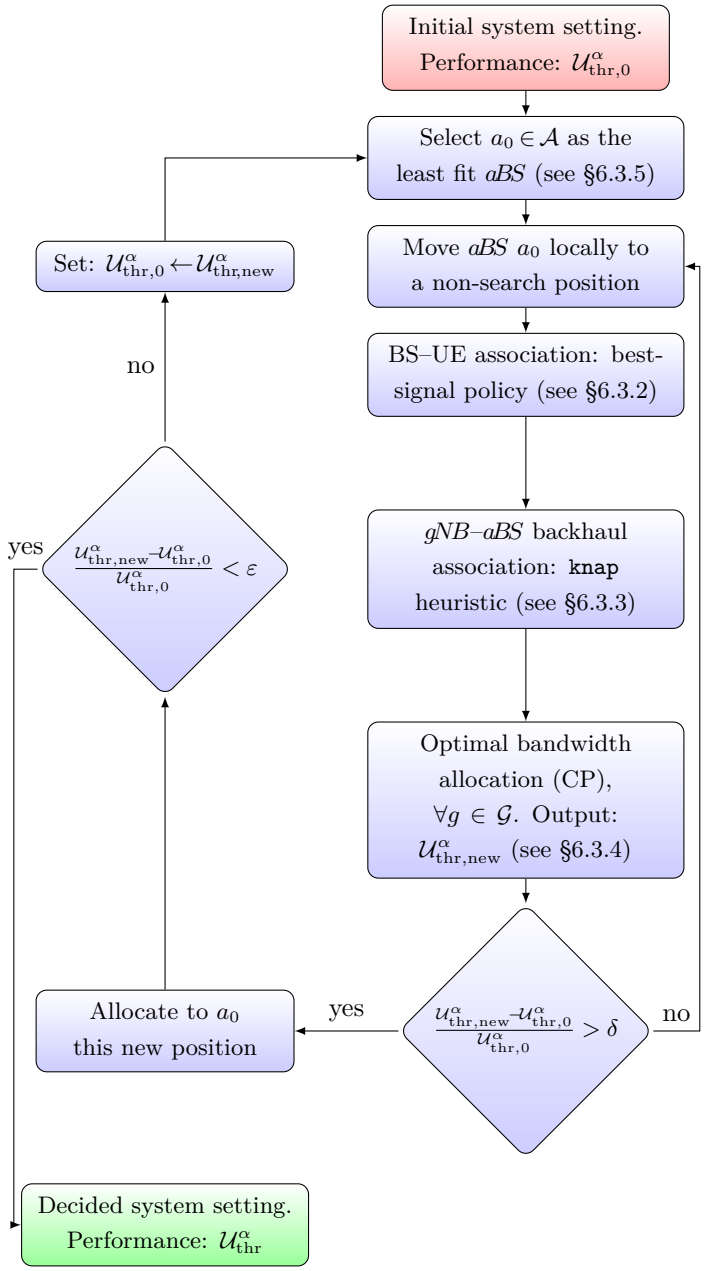


Figure 6.2: Flow diagram of `PADD` operation.

6.3.1. Initial System Setting

Initially, we consider a naïve drone positioning. For instance, any random placement of drones provides a feasible solution that could be iteratively improved. However, we more efficiently select those locations that are closer to *gNB*s –to guarantee good backhaul links–and those locations that are above densely populated regions–which are regions that potentially need drone relay assistance. This initial drone setting yields utility $U^{\alpha}_{\text{thr},0}$. However, in order to compute $U^{\alpha}_{\text{thr},0}$, we need to know also user association,

backhaul association and resource allocation. These decisions are made as described in the following subsections.

6.3.2. BS–UE Association: Best-Signal Policy

For fixed positions of the drones, the BS–UE association is performed individually by each user according to the best-signal policy described in Section 6.1.2, which is at most linear in the number of base stations $|\mathcal{B}|$, with $\mathcal{B} = \mathcal{G} \cup \mathcal{A}$. However, before association, we need to compute $|\mathcal{B}|$ SNR values for each UEs, and sort them in decreasing order, which goes with $|\mathcal{B}| \log |\mathcal{B}|$. The complexity of user association is therefore $\mathcal{O}(|\mathcal{B}| \log |\mathcal{B}|)$ comparisons, for each user.

In the following, the set of users attached to a *gNB* $g \in \mathcal{G}$ and the set of users attached to an *aBS* $a \in \mathcal{A}$ are denoted as $\mathcal{U}_g$ and $\mathcal{U}_a$, respectively.

6.3.3. *gNB–aBS* Backhaul Association: `Generalized Assignment Problem (GAP)-Knap` Heuristic

For fixed drone positions and given BS–UE association, the backhaul association is solved by assuming that the bandwidth $W_{\mathcal{G}}$ is proportionally shared among users connected to a *gNB* and to the *aBS*s to be connected to that *gNB*.

Specifically, the backhaul throughput is computed with the Shannon formula, using a fraction of bandwidth proportional to the number of users attached to the drone, and with the SINR resulting from current position of drone and the fixed position of *gNB*. Maximizing the α-fairness of such throughput values for all drones, translates into a GAP [149]. With the notation used in Eq. (6.2), in which $\gamma^{\mathcal{B}}_{g,a}$ is the backhaul SINR of link (b, a) and $B^{g,a}$ denotes the binary decision variable that tells wether *gNB* g associates with *aBS* a, we have:

$$\begin{cases} \max\limits_{B^{g,a}} U^{\alpha}_{\text{bhl}} = \begin{cases} \sum\limits_{g\in\mathcal{G}} \sum\limits_{a\in\mathcal{A}} \left(\frac{|\mathcal{U}_a|}{|\mathcal{U}_g \bigcup \mathcal{U}_a|} \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right) \right)^{1-\alpha} \cdot \frac{B^{g,a}}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{g\in\mathcal{G}} \sum\limits_{a\in\mathcal{A}} \log\left(\frac{|\mathcal{U}_a|}{|\mathcal{U}_g \bigcup \mathcal{U}_a|} \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right) \right) \cdot B^{g,a}, & \alpha = 1; \end{cases} \\ \text{s.t.:} \\ \sum\limits_{a\in\mathcal{A}} B^{g,a} \leq A_g, \qquad \forall g \in \mathcal{G}; \\ \sum\limits_{g\in\mathcal{G}} B^{g,a} = 1, \qquad \forall a \in \mathcal{A}. \end{cases} \tag{6.3}$$

The first constraint states that each *gNB* can provide backhaul service to at most A_g *aBS*s. The second constraint states that each *aBS* a must be associated with one *gNB*.

GAP is an NP-hard MILP [150]. Hence, although the GAP that `PADD` needs to solve have a small size and could be optimally solved by means of standard methods

as a Branch&Bound search [87], such an approach would not lead to a polynomial-time algorithm. Hence, multiple heuristics have been proposed in literature to find near-optimal solutions to the GAP [150, 151]. In particular, we perform a simple heuristic based on an approximation to the 0-1 knapsack problem [152] by means of dynamic programming [153]. We name this heuristic as `GAP-knap`, which has a polynomial complexity of $\mathcal{O}(|\mathcal{G}|\cdot|\mathcal{A}|\cdot A_g)$. The complexity of `GAP-knap` is linear with the sizes of $\mathcal{G}$ and $\mathcal{A}$ and the maximum number of *aBS*s allowed to attached to *gNB*s, A_g.

6.3.4. Optimal Bandwidth Allocation: Convex Program

For fixed drone positions, BS–UE association and *gNB*–*aBS* backhaul association, the optimal bandwidth allocation is solved by each *gNB* independently, in parallel, by means of a convex program.

Each *gNB* must perform bandwidth allocation in order to split backhaul resources among the served *aBS*s, split *gNB*–UE access resources among the set of attached users $\mathcal{U}_g$, and the attached *aBS*s a must split *aBS*–UE access resources among the set of users attached to each *aBS* a, $\mathcal{U}_a$. Since all these bandwidth allocations are intertwined (*aBS*–UE resource allocation depends on the bottleneck at the wireless backhaul link, *gNB*–*aBS* resource allocation depends on the number of attached *aBS*s to the same *gNB* and the number of final end-users of each of these *aBS*s, and *gNB*–UE access resources depend also on the backbone service that the *gNB* gets from the Internet), we formulate a convex program (6.4).

$$
\begin{cases}
\max\limits_{w^a, w_u, T_u} U^{\alpha}_{\mathrm{cvx},g} = \begin{cases} \sum\limits_{u \in \bigcup\limits_{b \in \mathcal{B}_g} \mathcal{U}_b} (T_u)^{1-\alpha} \cdot \frac{1}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{u \in \bigcup\limits_{b \in \mathcal{B}_g} \mathcal{U}_b} \log\left(T_u\right), & \alpha = 1; \end{cases} \\
\text{s.t.:} \\
w^a \geq W^{\min}_{\mathcal{B}}, & \forall a \in \mathcal{A}_g; \\
\sum\limits_{a \in \mathcal{A}_g} w^a = W_{\mathcal{B}}; & \\
T^a \leq w^a \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right), & \forall a \in \mathcal{A}_g; \\
w_u \geq W^{\min}_{\mathcal{G}}, & \forall u \in \mathcal{U}_g; \\
\sum\limits_{u \in \mathcal{U}_g} w_u = W_{\mathcal{G}}; & \\
w_u \geq W^{\min}_{\mathcal{A}}, & \forall u \in \bigcup\limits_{a \in \mathcal{A}_g} \mathcal{U}_a; \\
\sum\limits_{u \in \mathcal{U}_a} w_u = W_{\mathcal{A}}, & \forall a \in \mathcal{A}_g; \\
T_u \leq w_u \log_2(1+\gamma_{b,u}), & \forall (b,u) \in \mathcal{B}_g \times \bigcup\limits_{b' \in \mathcal{B}_g} \mathcal{U}_{b'} \mid u \in \mathcal{U}_b; \\
\sum\limits_{u \in \mathcal{U}_a} T_u \leq T^a, & \forall a \in \mathcal{A}_g; \\
\sum\limits_{u \in \mathcal{U}_g} T_u + \sum\limits_{a \in \mathcal{A}_g} T^a \leq \tau_g. &
\end{cases}
\tag{6.4}
$$

In the program, w^a is the share of total bandwidth that *gNB* g allocates to *aBS* a, T^a is the backhaul throughput for *aBS* a, w_u is the share bandwidth that BS b allocates to user u, and T_u is the access service throughput for user u. $\mathcal{A}_g$ is the set of *aBS*s attached to *gNB* g, and $\mathcal{B}_g = \{g\} \cup \mathcal{A}_g$ is the set of *gNB* g jointly with $\mathcal{A}_g$. $U^{\alpha}_{\mathrm{cvx},g}$ is the utility function of *gNB* g, which is based on the α-fairness metric.

The problem is convex, hence it is optimally solvable in polynomial time by means of standard interior-point methods [154]. However, these methods run in cubic time with respect to the number of users, i.e., the complexity is $\mathcal{O}(|\mathcal{U}|^3)$ [155]. This cubic complexity might soon become prohibitive for real-time applications such as drone-aiding and fast repositioning in wireless networks (specially for big populations), as addressed in this thesis. However, we have derived KKT conditions [156] for problem (6.4) that allow us to find the exact solution analytically (see Appendix F). The complexity of finding the exact solution for each *gNB* g is linear with respect to the number of users and number of *aBS*s attached to *gNB* g, i.e., $\mathcal{O}(U_{\max} \cdot A_g)$, in the worst case.

6.3.5. Least Fit Drone Selection

PADD iteratively improves the utility until convergence. The algorithm uses the idea behind EO algorithms. It selects the *least fit* element and re-sets its parameters in order to improve system performance. In our case, an *aBS* is selected and a new location is *probed*.

The choice of which *aBS* is the least fit is made based on the consideration that there are two factors that cause sub-optimality of *aBS* positions: (i) the *aBS* has a bad backhaul connectivity, hence it provides a worse service to users than what the access channel conditions allow, i.e., access resources are wasted; or (ii) the *aBS* offers bad access connectivity to users due to inter-drone interference, even though it has a good backhaul connectivity, so that backhaul resources are wasted. Accordingly, we derive two indicators of sub-optimality as the relative difference between the aggregate utility due to users connected to the *aBS* (denoted as $U^{\alpha}_{\text{thr},a}$) and the following quantities: (i) the utility of the *aBS* assuming infinite backhaul capacity, denoted as $U^{\alpha,B_\infty}_{\text{thr},a}$; and (ii) the utility the *aBS* assuming no inter-drone interference, denoted as $U^{\alpha,SNR}_{\text{thr},a}$. Eventually, we select as least fit *aBS* the drone a_0 that corresponds to the higher value of all sub-optimality indicators:

$$a_0 = \arg\max_{a\in\mathcal{A}} \left(\max \left(\left| \frac{U^{\alpha,B_\infty}_{\text{thr},a} - U^{\alpha}_{\text{thr},a}}{U^{\alpha,B_\infty}_{\text{thr},a}} \right|, \left| \frac{U^{\alpha,SNR}_{\text{thr},a} - U^{\alpha}_{\text{thr},a}}{U^{\alpha,SNR}_{\text{thr},a}} \right| \right) \right). \tag{6.5}$$

Computing utilities for *aBS* a has complexity $\mathcal{O}(U_{\max})$ sums and powers (logarithms for $\alpha = 1$). Hence, finding the maximum shown in Eq. (6.5) has complexity $\mathcal{O}(|\mathcal{A}| \cdot U_{\max})$ powers (or logarithms), the sums incurring negligible extra complexity.

Having identified a_0, PADD selects a new random position within the allowed 3-D ball space around the current position of the drone.

6.3.6. Overall Complexity of PADD

Our proposed algorithm consists of sequential steps, some of which involve operations that can be parallelized and can run on a centralized or distributed network orchestrator. Specifically, at each iteration, i.e., for fixed positions of drones and having identified the least fit drone, we have:

Step 1 *User association.* This can be implemented on parallel threads: one thread per UE computes and ranks the candidate list of BSs to attach to, then a separate thread computes the association in at most as many rounds as the number of BSs. With $|\mathcal{U}|$ parallel threads, the time required for this step goes with $|\mathcal{B}| \log |\mathcal{B}|$ sums and comparisons, as seen in Section 6.3.2.

Step 2 *Backhaul association.* The next step consists in solving the GAP-knap problem for backhaul association, which must be done with a single thread, with complexity

$\mathcal{O}(A_g|\mathcal{G}||\mathcal{A}|)$, as shown in Section 6.3.3.

Step 3 *Resource allocation.* This requires a thread per each *gNB* for which we need to solve the convex problem (6.4). The time needed to complete this step is therefore the time needed to solve a single problem, i.e., with complexity $\mathcal{O}(U_{\max} A_g)$, as shown in Section 6.3.4.

Step 4 *Utility evaluation.* The current configuration is evaluated in terms of α-fair utility (6.1), which has the cost of $|\mathcal{U}||\mathcal{B}|$ sums plus $|\mathcal{B}|$ power operations (for $\alpha \neq 1$) or logarithms (for $\alpha = 1$). As discussed at the beginning of this section and depicted in Figure 6.2, the current configuration can be discarded and a new position is probed for the current least fit drone (back to **Step 1**).[1]

Step 5 *Least fit selection.* Eventually, the algorithm has to compute the least fit drone with a single thread, with complexity $\mathcal{O}(|\mathcal{A}|U_{\max})$ powers (or logarithms, depending on α), as shown in Section 6.3.5.

With discretized drone positions, and indicating with N the number of points that can be probed within the ball of points that can be probed, the complexity of one iteration is N times the complexity of **Step 1**-**Step 4**, plus the complexity of **Step 5**. The overall complexity is therefore $\mathcal{O}(N(|\mathcal{B}| \log |\mathcal{B}| + A_g|\mathcal{G}||\mathcal{A}| + A_g U_{\max} + |\mathcal{U}||\mathcal{B}|))$ sums or comparisons and $\mathcal{O}(N|\mathcal{B}| + |\mathcal{A}|U_{\max})$ powers (or logarithms). This complexity is therefore low-degree polynomial, and linear in most of the parameters. The number of iterations required in EO algorithms is not bounded unless a hard limit is externally imposed, however, the approach is designed to quickly approach a local optimum.

In our experiments shown in the next section, we have observed no more than a hundred of iterations, without imposing any limit to the number of iterations. The number of possible testing positions for the least fit drone is $N = 400$ but we have observed numbers around five. Indeed, on average, the initial position can be improved with probability 0.5, which means that, if we assume that positions are selected at random with no memory, the iteration stops after testing, on average, 2 positions in the ball around the least fit drone (it would be a Bernoulli process). If we consider memory, and do not allow to probe the same position twice, the Bernoulli approximation yields an upper bound on the average number of probing attempts, because in reality the probability that the next position will be better than the current one, and the iteration will therefore stop, will grow attempt after attempt.

[1] When the optimization problem is initialized, there is no least fit drone yet; thus, the first iteration executes **Step 1**-**Step 4** only once, then it moves to **Step 5** to make the first least fit selection.

Figure 6.3: Topology of Leganés (Spain) and *gNB*s placement.

6.4. Numerical Simulations

Here we present the numerical evaluation of the α-fair cellular capacity optimization in both static and dynamic scenarios. All our simulations are performed over the real topology of an operational network deployed in a dense city: 150,000 inhabitants in a 10 km^2 area) (Leganés, south of Madrid, Spain). The area is covered by 10 *gNB*s using the same LTE band, as shown in Figure 6.3.

As there are several operators in Spain that provide LTE service over multiple bands, and we only consider one operator in one band, we use 1000 users as a reasonable number of users that request service at the same time, unless otherwise specified.

First, we numerically validate our proposed `PADD` in comparison with optimal results approximated by means of Monte-Carlo (`MC`) simulations in static networks with limited population. Second, we analyze the robustness of the system model in order to prove that assuming perfect knowledge about user positions has limited impact on performance. Eventually, we study three significant static and dynamic scenarios:

- *PPP*: we statically place UEs on the city map according to a Poisson point process [157].
- *Stadium*: we statically place 60% of the UEs in the surroundings of a stadium, and the rest like in *PPP*.
- *Event*: 40% of the population of UEs moves according to the random way-point model, whereas other UEs keep arriving at an official scheduled rate of a train station and move towards the stadium.

Table 6.1: Evaluation parameters

Parameter	***Value***
ξ_{LoS}, ξ_{NLoS}, β_1, β_2	1.6 dB, 23 dB, 12.08, 0.11
Carrier frequencies, $f_{\mathcal{G}}$, $f_{\mathcal{A}}$	1815.1 MHz, 2.63 GHz
Bandwidths, $W_{\mathcal{G}}$, $W_{\mathcal{A}}$	18 MHz, 18 MHz
Tx power, P^g_{Tx}, P^a_{Tx}	44 dBm, 25 dBm
Thermal Noise Power	-174 dBm/Hz
Ground path loss exponent, η_{BS}	3
Height range, $[h_{\min}, h_{\max}]$	$[40, 300]$ m
Urban area, $\lvert\mathcal{S}\rvert$	10 km^2
Average walking speed	2.5 m/s
Monte-Carlo runs per instance	10^7
Instances of simulations	1000

For each scenario, we study drone placement, utility and throughput achieved, and system fairness measured with the Jain's index [94], expressed as

$$J = \frac{\left(\sum\limits_{u\in\overline{\mathcal{U}}} \theta_u\right)^2}{|\overline{\mathcal{U}}| \cdot \sum\limits_{u\in\overline{\mathcal{U}}} \theta_u^2}, \tag{6.6}$$

where θ_u is the throughput of user $u \in \overline{\mathcal{U}}$.

Table 6.1 reports the evaluation parameters used in our simulations. We take a carrier bandwidth of 20 MHz for both *gNB* and *aBS* channels (out of which, we consider that 10% is for guard bands, so we only use 18 MHz [146]). The carrier frequency of cellular links is 1815.1 MHz, while for air-to-ground links we use 2630 MHz. These are two commonly used LTE bands. The transmission power of *aBS*s is 25 dBm, notably lower than the 44 dBm power transmission of the ground *gNB*s. In addition, probed aerial positions arise from a lattice that spans equal-volume subspaces.

To position our proposal, we consider the performance without drones as baseline (`Ground` in the figures). Moreover, to compare `PADD` to state of the art drone-position optimization frameworks, we have implemented and tested the ***`RA`*** [74] scheme.

Although our analysis holds for generic values of α, here we present results for three specific and most interesting cases:

(i) For $\alpha = 0$, we obtain the maximum throughput achievable (`MaxThr` in the figures).

(ii) For $\alpha = 1$, we optimize the proportional fairness network metric (`PropFair` in the figures).

(iii) For $\alpha \to \infty$, we optimize the max-min fairness network metric (`MaxMin` in the figures).

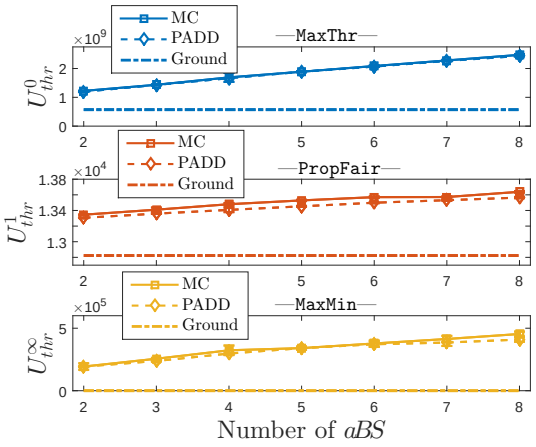


Figure 6.4: Utility validation for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *PPP*.

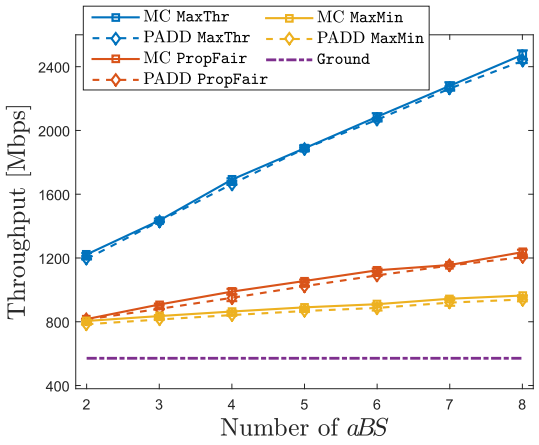


Figure 6.5: Network capacity validation for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *PPP*.

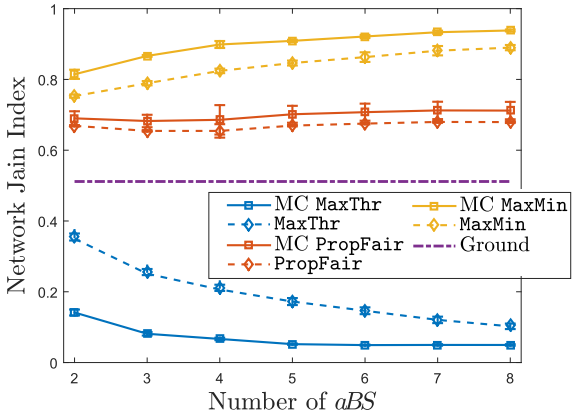


Figure 6.6: Network fairness validation for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *PPP*.

We study these three metrics for the following reasons. The `MaxThr` metric provides the maximum achievable network capacity, without fairness constraints. The `PropFair` metric takes into account the network capacity but it also does not let fairness decay. The mathematical design of such a metric, which optimizes the aggregation of logarithms of user throughputs, allows for finding a good trade-off between a high system capacity and reasonably high fairness values. The `MaxMin` metric only target fairness of the weakest customer, which comes at the cost of providing lower aggregate throughput. Both the `PropFair` and `MaxMin` have been adopted in the implementation of real telecommunication systems [158] as well as in many research works [159,160]. Complementary, in Appendix F, we provide more theoretical results for the `PADD` scheme for a generic value of α.

We have simulated every analyzed use-case 1000 times using MATLAB `R2020a` and show average results. Error bars in the figures are 95% confidence intervals.

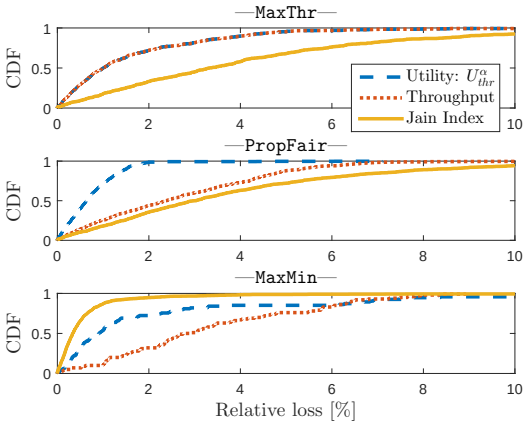


Figure 6.7: Robustness validation for $\alpha \in \{0, 1, \infty\}$. CDF of the relative loss. $G = 10$, $A = 5$, $U = 1000$. Scenario: *PPP*.

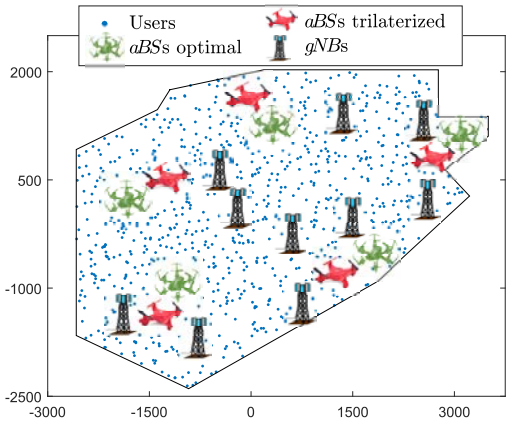


Figure 6.8: Robustness validation for $\alpha = 1$. *aBS*s placement error. $G = 10$, $A = 5$, $U = 1000$. Scenario: *PPP*.

6.4.1. Validation of PADD Operation

Here we compare PADD results and optima (approximated by means of MC, since there are no computationally feasible alternatives) in the *PPP* scenario, and comment on the basic properties of our approach.

In Figure 6.4, we present a utility comparison between the Ground scheme, optimum results from MC and our proposal PADD with the three main metrics studied in this chapter (MaxThr, PropFair and MaxMin). Note that different values of α make utility values very different, so we cannot compare utilities across fairness metrics. So, we will compare PADD and the existing schemes on a per-metric basis.

We observe that the difference between utilities achieved with PADD and MC is negligible, below 1% in all cases. This shows that our proposal PADD is able to achieve close-to-optimal results with a much lower complexity. Furthermore, PADD matches very well also the throughput achievable in the optimal case, as illustrated in Figure 6.5. This shows that PADD seeks the optimal system configuration, and not just an operational configuration that is near-optimal according to the chosen utility metric.

As concerns Jain's fairness, Figure 6.6 shows interesting patterns. With PropFair and MaxMin, PADD is slightly below MC (5% in the worst case). This happens because the Jain's index does not match the definitions of proportional fairness and max-min fairness. The key difference is that the Jain's index penalizes any unbalance with a quadratic symmetric function, while the other two metrics are strongly non-linear and asymmetric. Instead, with MaxThr, PADD obtains fairness values above the ones of MC. This is due to the fact that PADD is parallelized and does not find the exact configuration that maximizes throughput, while MC does. Paradoxically, this turns into less imbalanced throughput in the case of using PADD. Appendix F provides more insights on the matter.

The figures show that, in the PPP scenario, utility and throughput increase with fleet size. This is because user density is homogeneously spread over the entire area, so that drones are spaced apart, and inter-drone interference constraints do not kick in. Later we will see that this is not always the case. Note also that the gain in terms of network utility and throughput with respect to the `Ground` configuration is remarkable under all considered metrics, which confirms that coordinated drone relays have huge potential.

The figures also show that fairness does not necessarily increases with the number of drones, and can even be worst than without drones, when the target is pure throughput maximization. However, with fairness-concerned metrics like `PropFair` and `MaxMin`, the use of drones offers clear advantages.

6.4.2. Robustness of `PADD`

In the system model described in Section 6.1, we have assumed that the positions of users are known. However, it might be not realistic to estimate user locations with negligible error unless GPS is enabled or many drones are available [71]. For instance, using a trilateration on the signal strength at the base station, the error on the position is normally below 50 m [161]. Hence, here we consider the PPP scenario to numerically analyze the robustness of `PADD` by introducing uncertainty in the position of the ground users, uniformly at random, within 50 m.

Figure 6.7 depicts the CDF of the relative loss due to erroneous user position estimation, i.e., the relative loss of utility due to optimizing drone positions according to erroneous user positions. The figure also shows the error in terms of throughput and fairness separately. The loss is below 10% in all cases, and below 3% for utility and throughput in more than half of the cases, while the average loss is below 5%. Indeed, the position of drones selected by `PADD` is similar with and without localization errors (see Figure 6.8). The reason of such robust behavior stands in the fact that the optimization of drone positions is done based on many users and in relatively large areas, so that multiple errors in user positions are not so important, whereas the presence of a distributed mass of users in a given area is what actually catalyzes the presence of a drone.

Next, we analyze non-homogeneous user topologies in static and dynamic cases.

6.4.3. Performance Evaluation in the Static *Stadium* Case

The *Stadium* scenario allows us to study network performance when the ground network cannot sustain the user demand.

In Figure 6.9 we show the average performance experienced by all users in the scenario, while in Figure 6.10 we focus on the performance of users by the stadium. We observe that `PADD` with `MaxThr` benefits of the presence of drones (see Figure 6.9, top), although adding drones is negative for users by the stadium (see Figure 6.10, top). In fact, adding *aBS*s

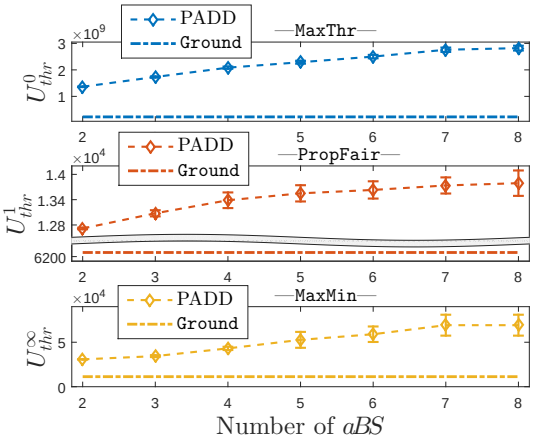


Figure 6.9: Utility of all users for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *Stadium* with $U_d = 600$.

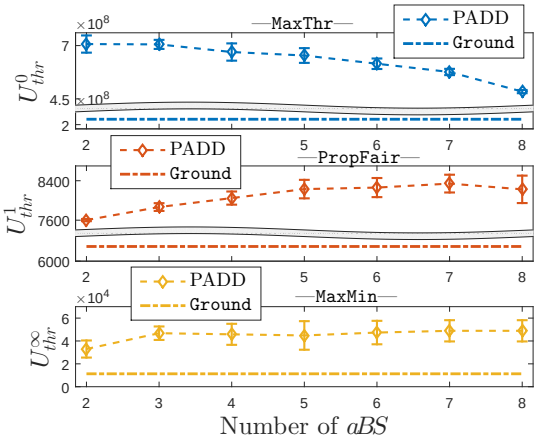


Figure 6.10: Utility of stadium users for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *Stadium* with $U_d = 600$.

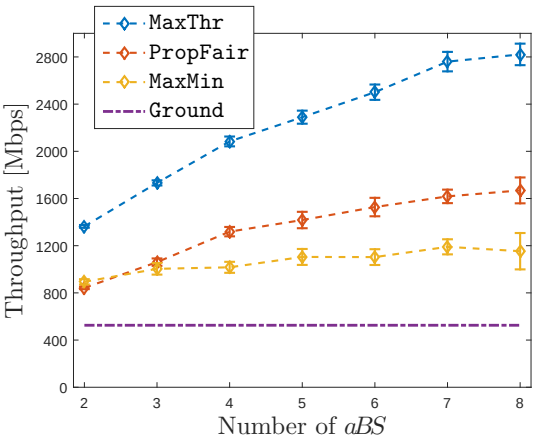


Figure 6.11: Throughput of all users for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *Stadium* with $U_d = 600$.

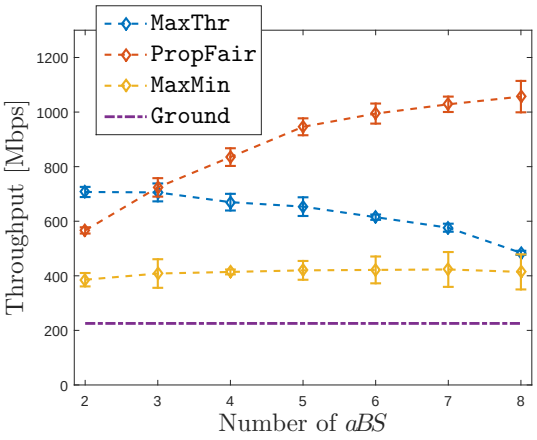


Figure 6.12: Throughput of stadium users for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $U = 1000$. Scenario: *Stadium* with $U_d = 600$.

introduces more capacity and connectivity opportunities, hence the benefit at aggregate city level. However, `PADD` with `MaxThr` seeks for drone positions where they are less impaired by interference, so to generate a few good-quality channels. This behavior results into having only one or two *aBS*s at most located near the stadium. More *aBS*s would interfere too much. Hence, as long as *aBS*s are added, `PADD` with `MaxThr` positions them apart from the stadium, yet they generate some interference, which progressively worsens the performance of users by the stadium.

`PADD` with `PropFair` behaves completely different. It brings drones where it to favors users otherwise served below the average, so to increase the system's log utility. This results in deploying *aBS*s by the stadium, where the density of users in presence of limited radio resources hinders performance more than radio channel quality issues.

With `MaxMin`, `PADD` positions drones where users would otherwise suffer the worst connection quality, irrespective of their closeness to densely populated areas. Therefore the aggregated throughput is lower than with `PropFair`, which in turn is much lower than with `MaxThr`, as shown in Figure 6.11.

If we now consider only users by the stadium, Figure 6.12 illustrates how using `PropFair` can clearly outperform `MaxThr` in terms of throughput. This is due to the fact that the optimization of throughput requires positioning drones where they interfere less, which is not necessarily by the stadium. Indeed, due to interference, users by the stadium could incur a loss by increasing the number of drones even in the case of using `MaxMin`, as shown in the figure.

As a result of the previous numerical analysis, we observe that using *aBS*s can be beneficial to help the ground cellular network in dense spots, except it cannot help in purely maximizing throughput (e.g., with `MaxThr`). Drones cannot either "rescue" all users with bad channel conditions, as `PADD` would seek with `MaxMin` and `PropFair`. However, `PADD` can always provide fairness and large throughput gain with respect to the `Ground` case. Besides, the version of `PADD` with `PropFair` results to be quite effective in case of dense masses of users.

6.4.4. Performance Evaluation in the Dynamic *Event* Case

The last scenario we consider is dynamic and allows us to study the evolution of network performance *while* the density of users increases. Moreover, it shows the importance of designing a fast and reactive algorithm to re-position drones as the user topology changes.

In the Event scenario, small masses of 40 users arrive periodically to the train station of the city every 5 minutes. The initial population is 400 users and keeps growing during 75 minutes up to 1000 users. There are 5 *aBS*s hovering the area in this example. Upon a train arrival, the new mobile users walk towards the stadium, located 1.5 km away from the train station. The fleet is repositioned every 5 simulated minutes, using as initial condition the positions of the 5 drones in the previous optimization epoch.

We first illustrate how the network evolves over time in Figures 6.13 and 6.14, using `PADD` with `PropFair`. After 25 minutes (Figure 6.13) some people are already at the stadium, while many others keep arriving and are walk towards it. At that point in time, 3 *aBS*s from the fleet of drone relays are getting prepared to serve the users nearby the stadium and also the smaller groups on their way from the train station to the stadium. We see how drones adapt their positions after other 35 minutes, in Figure 6.14, when much more users have arrived at the stadium. By that time, one more drone has been dispatched as well, to serve the stadium. The trajectories of the drones for a 75-minute simulation instance are shown in Figure 6.15, where diamonds represent the source position of *aBS*s. In the figure, we can see that *aBS* 5 is not required to assist the dense stadium spot, while

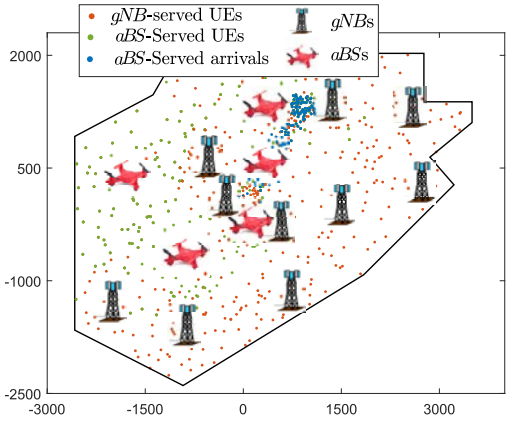


Figure 6.13: Network state in $t = 25$ min. $G = 10$, $A = 5$, $U = 640$. Scenario: `PropFair`, *Event*.

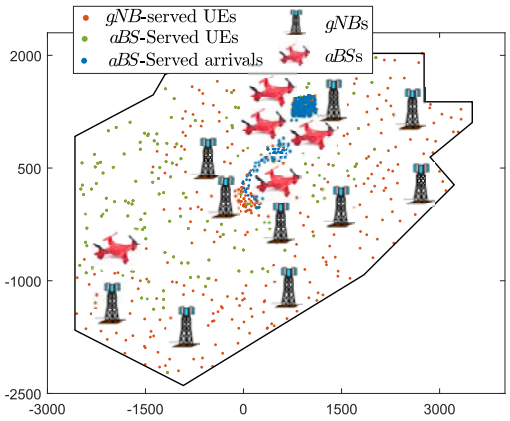


Figure 6.14: Network state in $t = 60$ min. $G = 10$, $A = 5$, $U = 880$. Scenario: `PropFair`, *Event*.

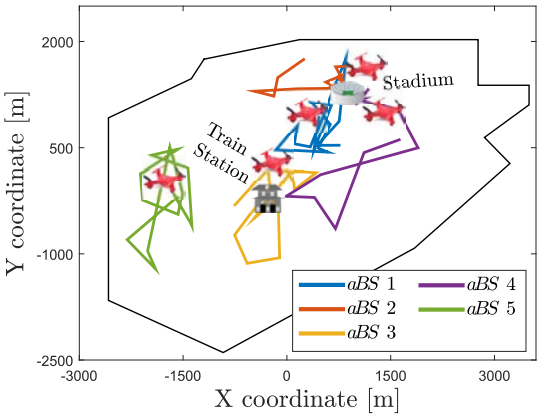


Figure 6.15: Drone trajectories during 75 minutes. $G = 10$, $A = 5$, $U = [400, \ldots, 1000]$. Scenario: `PropFair`, *Event*.

*aBS*s 1 and 2 are always hovering between the train station and the stadium. Also, *aBS*s 3 and 4 keep moving back and forth within different regions, in order to fairly supply the demand of users.

As concerns performance, Figures 6.16 and 6.17 illustrate utility and throughput as they evolve over time. Here, in addition to `PADD` with the three selected values of α, we also compare the `RA` scheme. With `RA`, *aBS*s are attracted by UE's inverse SNR, and repulsed by proximity to *gNB*s to avoid interference. `RA` does not target any specific fairness metric, so we quantify its impact with the same utility functions used for `PADD`, computed for users arriving over time (the *attendance*).

As time passes by, and more people reach the stadium, Figure 6.16 shows a significant utility raise under the adoption of either `MaxThr` or `PropFair` schemes. With `MaxThr`, the gain of `PADD` over `RA` and `Ground` schemes is high, although it saturates quickly. Instead,

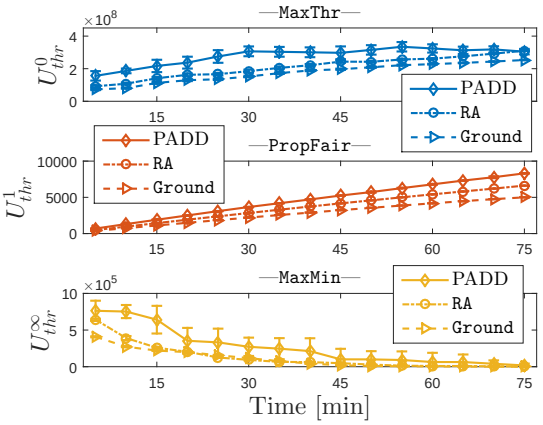


Figure 6.16: Attendance utility for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $A = 5$, $U = [400, \ldots, 1000]$. Scenario: *Event*.

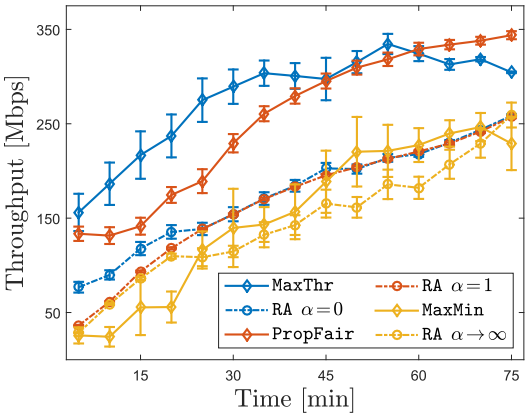


Figure 6.17: Attendance throughput for $\alpha \in \{0, 1, \infty\}$. $G = 10$, $A = 5$, $U = [400, \ldots, 1000]$. Scenario: *Event*.

with `PropFair`, `PADD` exhibits a smoother behavior, as its gain keeps increasing.

Under the `MaxMin` scheme, `PADD` performs better than `RA`, although now we observe a decay of performance over time, for all schemes. This is due to the fact that, with more attendance, the minimum per-user achieved rate will decrease, unless more drones were deployed.

In all cases, the aggregate throughput experienced by the users keeps increasing for all the schemes and all the values of α, as depicted in Figure 6.17. In particular, `PropFair` exhibits a similar—slightly better in some of the cases—increase than `MaxThr`. This is in accordance with the results commented before for the users at the stadium in the static analysis. Clearly, the `RA` scheme is not able to opportunistically take advantage of user diversity and improve utility because it does not target a throughput-based metric, unlike `PADD`. This permits our scheme to optimize the network with much better guarantees in terms of throughput and fairness.

6.5. Lessons Learnt and Discussion

The performance assessment carried in this chapter shows the importance of integrating a fleet of drone relay stations in a cellular network. It also unveils that optimizing drone positions to maximize throughput, without taking into account fairness, has little relevance in the presence of dense spots or ground users. Instead, a fair metric like `PropFair` provides notable throughput and utility improvements.

The fact that `PADD` is efficient, makes it possible to design an almost continuous reconfiguration of positions in realistic networks. In turn, our scheme is fast because we have designed it by segmenting the problem to solve into a few phases: we use EO for optimizing drone positions, while wireless backhaul attachment, BS selection and

distributed resource allocation are sequentially solved optimally and analytically for each drone topology.

The actual implementation of PADD requires the exchange of signaling information between drones and a centralized orchestrator, as well as the implementation of a mechanism to track users. Signaling incurs some limited overhead to instruct drones and to gather user positions and interference reports (which are however already collected by current base stations), depending on the frequency of reconfigurations. However, we have seen that PADD is robust to imprecise tracking of user positions, which can be therefore strongly simplified, so that the additional overhead due to PADD will mostly be due to controlling drones.

Finally, we comment on drone mechanicals. Current commercial drones can carry small BSs and access points, although they cannot fly for long time due to battery limitations (around 30 minutes at most). They can easily move at the reasonable speed of 15 m/s. Therefore, it is possible to derive a repositioning scheme that accounts for replacing drones that go back to the charging station if the drones do not fly too far from it. For the case considered in Section 6.4.4, the routes flown by drones in 5 minutes are long enough, and it is easy to hover a city district in a few minutes. Thus, notwithstanding the intricacies of the analysis, the performance evaluation discussed in this chapter is quite relevant for realistic systems.

7 Conclusions and Final Remarks

In this thesis, we have addressed and contributed to the design of solutions that cope with the need of flexible and adaptive relaying strategies in wireless networks. These solutions allow novel communications paradigms to perform efficiently and effectively in the advent of the 5G and beyond networks. Relaying techniques manifest great opportunities for switching to fastest traffic paths opportunistically, provide improved coverage and fair service rates and notably speed up the readiness of files at edge nodes. Hence, we have presented a compendium of optimization tools that boost end-user performance in key paradigms of wireless cellular networks.

The derivation of such optimization tools bring unavoidable challenges. Optimizing wireless performance under relay alternatives necessarily needs to model equations and constraints with integer variables. This fact translates into proposing optimization programs that easily turn into NP-Complete problems. Jointly with large sets of variables, coming from the highly mutable density of users, such optimizations are nowadays impossible to solve in efficient time. However, telecommunication operating systems require solutions in a millisecond scale. Hence, it has been key to deeply study the intrinsicalities of relaying problems and find approximation algorithms that find close-to-optimal solutions in efficient time.

Initially, we have studied the control in adaptive multi-mode D2D relay communications in cellular networks. We have proposed Multi-Path D2D (MPD2D), that stands as a D2D optimization framework in which D2D modes are adaptively selected with flows split over multiple D2D paths. Results obtained show an extremely high gain in terms of throughput in comparison to state of the art schemes facing the D2D mode selection problem. The use of multiple paths has been the first key contribution of this research. Although it comes with several technology restrictions, yet it shows significant potential gain that it is worth exploring. In addition, we have derived a system analysis framework and in particular we have introduced a flow satisfaction metric. Such metric provides memory to the system to make fairer link allocation decisions according to the proportional fairness paradigm, with no bias for freshly arrived flows and for

short-lived flows. By defining a new filtering technique, namely Dynamic Exponential Moving Average (DEMA), we have reduced to the minimum the implementation costs of fast reactive memory-enabled mode selection decisions in dynamic contexts. To further reduce complexity, we have proposed *D2D Intensive Multi-mode Multi-path (`DIMM`)* and *D2D Expeditious Multi-mode Multi-path (`DEMM`)*. These two heuristics achieve close-to-optimum results and carry out mode selection in a way that largely outperforms state of the art solutions.

As an alternative network-type static relay with novel novel Radio Access Technologies (RATs) such as Millimeter-Wave (mmWave), we have defined, analyzed and solved the mmWave Backhaul Scheduling (MMWBS) problem. This problem addresses compact and efficient scheduling of relaying in mmWave backhauls using an MILP formulation. We have proved that solving the problem is NP-hard because mmWave links incur a non-zero activation cost, due to antenna steering and signalling messages exchange typical of mmWave. We have derived and studied tight bounds for the length of the scheduling for a given set of download jobs, i.e., the *makespan*. We have shown that MMWBS cannot be approximated unless the interference can be neglected. In such case, based on linear programming, we are able to characterize the MMWBS problem analytically with constant-approximation guarantees using a scheduler running in polynomial time. We have also proposed practical heuristics, namely `Resched` and `Greedy`. We have evaluated the heuristics and validated the analysis by means of numerical simulations and, for small instances of the problem, compared heuristics to the optimal schedule computed with a Branch&Bound solver. Our results show that simple heuristics show notable gains in small testable systems. On average, these heuristics find near-optimal solutions under different network topologies and base station settings, both with and without the effect of interference between transmitting mmWave links.

A study of MMWBS under other interference models is left for future work. For instance, SINR [162], link-to-link– [163] and node-to-link affectance [164,165], and conflict graphs [100,101]. Also, other objective functions, such as minimizing energy consumption or optimizing beamsteering (since we consider those procedures described in current standards) can be further analyzed in the future. Questions of how to achieve stability, low latency, or high throughput under adversarial or stochastic packet injections are also a topic of interest to investigate.

Finally, we have moved from user- and network-type relays to extremely-mobile relay scenarios powered by aerial drone relays. We have proposed novel optimization frameworks for drone-aided cellular networks in terms of user coverage under guaranteed signal quality. Also, in terms of an α-fair throughput utility function under realistic stochastic models. We have addressed multiple coordinated drones as well as legacy *gNB*s and analytically accounted for complex interference expressions caused by drone transmissions under non-negligible and variable LoS probability. Specifically, we have

studied the integration of a finite fleet of drones acting as *aerial base stations* connected to and aiding a ground network of *gNB*s, from/to which they are able to relay traffic by means of 3D-beamforming wireless backhaul connections. Since cellular users can move, we have shown that implementing a solution for our frameworks requires solving three important subproblems: finding optimal positions of drones at a given time instant, map drones onto best-performance identified points, and plan flight routes. We have presented the coverage problem $\mathcal{C}$ and the throughput problem $\mathcal{T}$. Separately, both optimization problem are a non-linear, non-convex and mixed-integer NP-Complete problem, so it is not possible to handle them with conventional off-the-shelf optimizers. The main issue in the formulation lies on the intertwined nature of interference, drone positioning, and LoS probability. Hence, we have accordingly proposed `OnDrone` and `PADD`, two *extremal-optimization*-based algorithms that perform near-optimally in low-order polynomial time for various user topologies and under different density of LoS obstructions, from *suburban* to *high-rise* scenarios. `OnDrone` has shown to outperform state of the art mechanisms because they do not address the root causes of interference. Interestingly, we have found out that unnecessarily large drone fleets would only have negative impacts on coverage, due to interference. Besides, we have unveiled that optimally mapping drones onto coverage targets is doable in negligible time and, most importantly, we have introduced and evaluated a novel and dynamic scheme to compute intelligent drone trajectories upon repositioning. With our *Bézier Scheme*, we have shown that it is possible to deflect flight routes of drones so to dramatically improve coverage performance in a variety of scenarios, including in real, dense and high-populated cities. `PADD` leverages on parallel threads operations and provides near-optimal solutions in low-degree polynomial time, with a linear dependency on all system parameters but for the number of base station, which causes a sub-quadratic dependency. This makes `PADD` suitable for implementation in dynamically changing environments. The performance evaluation presented has shown that `PADD` brings significant gain and outperforms existing approaches. It also unveils that using fairness is key to get benefit from coordinated yet interfering drone relay stations.

Appendices

A Pareto-Optimality of MPD2D

Given the node utility defined in Eq. (3.6), it is easy to observe that parameters and variables indexation and notations can be re-adjusted so to express Eq. (3.6) as:

$$U_n = \sum_{i \in I} (\theta_n^i - \alpha_s E_n^i) \cdot Y_n^i + \sum_{j \in J} (\theta_n^j - \alpha_s E_n^j) \cdot Y_n^j, \tag{A.1}$$

where I and J are sets of indices. Hence, index sets I and J can be joined in $K = I \cup J$ and we can express Eq. (A.1) as:

$$U_n = \sum_{k \in K} (\theta_n^k - \alpha_s E_n^k) \cdot Y_n^k = \sum_{k \in K} \theta_n^k \cdot Y_n^k - \alpha_s \sum_{k \in K} E_n^k \cdot Y_n^k. \tag{A.2}$$

Now, as the system utility accounts for the summation of all node utilities, we have that:

$$U_{net} = \sum_{n \in \mathcal{N}^*} U_n = \sum_{n \in \mathcal{N}^*} \left(\sum_{k \in K} \theta_n^k \cdot Y_n^k - \alpha_s \sum_{k \in K} E_n^k \cdot Y_n^k \right). \tag{A.3}$$

Again, if we re-adjust the indexation, we have that the system utility is a combination of a throughput decision variable and an energy decision variable:

$$U_{net} = \sum_{l \in L} \theta_l \cdot Y_l - \alpha_s \sum_{l \in L} E_l \cdot Y_l = \sum_{l \in L} \theta_l^Y - \alpha_s \sum_{l \in L} E_l^Y, \tag{A.4}$$

where $L = \mathcal{N}^* \cup K$, $\theta_l^Y = \theta_l \cdot Y_l$ and $E_l^Y = E_l \cdot Y_l$.

Now, denoting by $\mathbf{Y}$ the set of binary decision variables, and $\theta(\mathbf{Y})$ and $E(\mathbf{Y})$ the throughput and energy functions depending on the binary decision variables, the network utility may be expressed as:

$$U_{net} = \theta(\mathbf{Y}) - \alpha_{\mathbf{s}} \mathbf{E}(\mathbf{Y}). \tag{A.5}$$

So far we have defined the system utility (U_{net}) as a combination of a throughput function ($\theta(\mathbf{Y})$) and an energy consumption function ($E(\mathbf{Y})$). In reality, what we aim in our framework is to solve a multi-variable optimization problem in which we maximize users throughput and minimize energy consumption. Hence, we should maximize the tuple $(\theta(\mathbf{Y}), -\mathbf{E}(\mathbf{Y}))$, i.e., we need to find those binary decision variables $\mathbf{Y}$ such that $\theta(\mathbf{Y})$ cannot be increased without increasing $E(\mathbf{Y}))$ and $E(\mathbf{Y}))$ cannot be decreased without decreasing $\theta(\mathbf{Y})$. This is a Pareto-optimal solution [85].

As we are indeed maximizing $U_{net} = \theta(\mathbf{Y}) - \alpha_{\mathbf{s}}\mathbf{E}(\mathbf{Y})$, we need to prove that the solution $\mathbf{Y}$ found in this way is Pareto-optimal. As linear combination of the multiple functions is the common way of finding Pareto-optimal solutions (as done, e.g., in [36]), the proof of the following lemma is quite simple.

Lemma 3. *The solution found in an optimization problem with* $U_{net} = \theta(\mathbf{Y}) - \alpha_{\mathbf{s}}\mathbf{E}(\mathbf{Y})$ *as the utility function corresponds to the Pareto-optimal solution of the multi-objective optimization problem that aims to optimize the tuple* $(\theta(\mathbf{Y}), -\mathbf{E}(\mathbf{Y}))$.

Proof: Let $\mathbf{Y}^*$ be the optimal solution of the single-utility optimization problem. Assume, *reduction ad absurdum*, that there is an alternative solution $\widetilde{\mathbf{Y}}$ that is the Pareto-optimal solution of the multi-objective optimization problem but it is not the optimal solution of the single-objective optimization problem. Hence, we have that:

$$\begin{aligned} \theta(\widetilde{\mathbf{Y}}) &\geq \theta(\mathbf{Y}^*), \\ -E(\widetilde{\mathbf{Y}}) &\geq -\mathbf{E}(\mathbf{Y}^*), \end{aligned} \tag{A.6}$$

where at least one inequality is not an equality. Hence, $\theta(\widetilde{\mathbf{Y}}) - \alpha_{\mathbf{s}}\mathbf{E}(\widetilde{\mathbf{Y}}) > \theta(\mathbf{Y}^*) - \alpha_{\mathbf{s}}\mathbf{E}(\mathbf{Y}^*)$.

Since, $\mathbf{Y}^*$ is the optimal solution of the single-utility optimization problem, we have a contradiction and the claim follows. ■

B From Non-Linear Optimization to MILP in MPD2D

The MPD2D optimization program derived in section 3.3.3 is non-linear because it has one minimum-like constraint and quadratic constraints. Both cases can be linearised by standard methods, as described in what follows.

Given a user $n \in \mathcal{N}$, the constraint:

$$Y_n^3 = \min\left(1, \sum_{m \,|\, (n,m)\in\mathcal{L}} Y_{n,m}^3 + \sum_{m \,|\, (m,n)\in\mathcal{L}} Y_{m,n}^3\right) \tag{B.1}$$

can be expressed in a linear way by means of imposing the following two constraints:

$$\begin{aligned} 2 \cdot Y_n^3 &\geq \sum_{m \,|\, (n,m)\in\mathcal{L}} Y_{n,m}^3 + \sum_{m \,|\, (m,n)\in\mathcal{L}} Y_{m,n}^3; \\ 2 \cdot Y_n^3 &\leq 1 + \sum_{m \,|\, (n,m)\in\mathcal{L}} Y_{n,m}^3 + \sum_{m \,|\, (m,n)\in\mathcal{L}} Y_{m,n}^3; \end{aligned} \tag{B.2}$$

given that $Y_n^3 \in \{0,1\}$ is a binary variable.

Clearly, constraints in Eq. (B.2) are equivalent to the minimum-like constraint in Eq. (B.1) because each of the summations can take only values 0 or 1 (according to the rest of the constraints of the optimization program). Hence, the equivalency follows.

Regarding the quadratic constraints, we have always the case in which the constraint contains the product of two binary variables: $Y_{n,m}^i$ and $Y_{n',m'}^{i'}$. In general, the product of two binary variables $x, y \in \{0,1\}$ can be linearised by means of introducing an auxiliary binary variable $\pi \in \{0,1\}$ and the following three linear constraints:

$$\begin{aligned} \pi &\leq x; \quad \pi \leq y; \\ \pi &\geq x + y - 1. \end{aligned} \tag{B.3}$$

Now, the binary variable π contains the value of the product of x and y, i.e., $\pi = x \cdot y$. Thus, we can replace any product of binary variables of the optimization program by an

auxiliary variable that accomplishes the linear constraints of Eq. (B.3).

The transformations described in this section turn the non-linear optimization program of Section 3.3.3 into an MILP. Hence, we can apply efficient and fast algorithms to solve the MILP, which is less complex than the non-linear program.

C Proof of NP-Completeness of the Coverage Problem $\mathcal{C}$

Theorem 5. *The Coverage Problem $\mathcal{C}$ is NP-Complete.*

Proof: We claim that the coverage problem $\mathcal{C}$ is NP-Complete, since the *MGDC* problem can be reduced, in polynomial time, to a particular case of the Coverage Problem $\mathcal{C}$. As the *MGDC* problem is a well known NP-Complete problem [125], then $\mathcal{C}$ is also NP-Complete. We prove such claim in what follows.

From the statement of the *MGDC* problem, we have a set of points $\mathcal{U}$ (users in our case) and a fixed radius $R>0$. Thus we want to minimize the number of disks with radius R that cover all the points of $\mathcal{U}$. We consider a specific set of instances of the Coverage Problem $\mathcal{C}$ in which the interference is negligible, all drones hover at the same elevation and are fast enough to reach any position in the surface (i.e., $\mathcal{S}_d = \mathcal{S}$). In this way, the ground coverage area of each *aBS* is a circle with a radius that depends on the elevation. Let's use elevation h_R at which the ground coverage area has the radius R of the *MGDC*. Thus, we show that this is a particular set of instances of the coverage problem that cannot be deterministically solved in polynomial time.

Assume, *reductio ad absurdum*, that problem $\mathcal{C}$ is solvable in polynomial time. for all its instances. Then, we present in Algorithm 8 an algorithm that solves any instance of the *MGDC* problem in polynomial time. Specifically, the *MGDC* problem that finds the minimal number of disks of radius R that cover all points in a set of 2–D coordinates can be solved by a linear search of the minimum number of disks. The search proceeds by adding a disk per iteration. It verifies that a given number of disks covers all points by solving the Coverage Problem $\mathcal{C}$ for the special instances described above. If the solution of the Coverage Problem $\mathcal{C}$ covers all points, the algorithm stops since we have found the minimum number of disks that cover all points. Note that in Algorithm 8, the iteration counter d (which is also the number of disks to use in one iteration) is bounded by the number of points $|\mathcal{U}|$. Thus, Algorithm 8 has a polynomial complexity, since we are assuming that problem $\mathcal{C}$ is solvable in polynomial time, which is a contradiction with the fact that the *MGDC* problem is NP-Complete. ■

D Bézier Curves for Drone Flight Paths

Bézier curves are an outstanding geometrical tool that deflects the trajectory of an *aBS* close to clusters of UEs while targeting the optimum positioning. *Bézier curves* are smooth, endpoint interpolators—i.e., the curve begins and ends in a provided source and destination—and are contained in the convex hull of a set of *anchor points*. The curve is *attracted* by *anchor points*, which in our case are specific users. Below, we mathematically define a *Bézier curve*.

Definition 1. *Given a set of points* $\mathcal{P}=\{P_k\}_{k=0}^{n}\subset\mathbb{R}^m$*, the resulting Bézier curve is:*

$$\beta^{\mathcal{P}}(t)=\sum_{k=0}^{n}\binom{n}{k}\cdot P_k\cdot t^k\cdot(1-t)^{n-k},\quad t\in[0,1]. \tag{D.1}$$

From the definition, we see that the *Bézier curve* $\beta^{\mathcal{P}}(t)$ begins at the source $\beta^{\mathcal{P}}(t=0)=P_0$, and ends at the destination $\beta^{\mathcal{P}}(t=1)=P_n$. The variable range for variable t can be adapted with a linear transformation to modify the speed of the curve. Since we are interested only in the flight path, i.e., the trajectory of the curve, we consider $t\in[0,1]$.

We need to obtain the offset region of the *Bézier curve* (a.k.a. stroking the curve). Since *Bézier* offset curves are not analytically obtainable [166], we use the *de Casteljau* algorithm [79]. This is the main recursive and numerically stable method to draw *Bézier curves* from a set of anchor points $\mathcal{P}$, by approximating the curve with small segments. We use *Bézier curves* for the ground projection of the trajectories, while the elevation of drones varies linearly from source to destination heights, indicated as h_{src} and h_{dest}. The actual 3–D trajectory of a drone with anchor points $\mathcal{P}$ is:

$$\vartheta^{\mathcal{P}}(t)=\left(\beta^{\mathcal{P}}(t),\ \ h_{src}+t\cdot(h_{dest}-h_{src})\right),\quad t\in[0,1]. \tag{D.2}$$

Finally, we stroke $\beta^{\mathcal{P}}(t)$ by stroking the segments obtained by the *de Casteljau* algorithm.

Algorithm 8 Solution for the *Minimum geometric disk cover* problem based on the solution of the Coverage Problem $\mathcal{C}$.

Input: Set of points $\mathcal{U}$, radius $R>0$.

1: $d \leftarrow 1$.
2: `found` $\leftarrow$ False.
3: **while** not `found`
4: Solve problem $\mathcal{C}$ for the set $\mathcal{U}$ and fixed elevation h_R.
5: **if** problem $\mathcal{C}$ covers all points $\mathcal{U}$, **then** `found` $\leftarrow$ True.
6: $d \leftarrow d+1$.
7: **end of while**

E Notes on OnDrone Operation

E.1. Guaranteed User Data Rate

The coverage optimization problem addressed in Chapter 5 aims to maximize the amount of ground users that are covered, either by *gNB*s or by *aBS*s, with a guaranteed service availability that allows for a minimum user data rate. Hence, as soon as we determine parameters as the maximum number of users U_g and U_d allowed to attach to a *gNB* g or *aBS* d, respectively, and the data rates $R^{\mathcal{A}}_{\min}$ and $R^{\mathcal{B}}_{\min}$, there is a user data rate guaranteed for each covered user. In Chapter 5, we mainly use an SINR value of 10.9 dB because it allows to use a 16QAM MCS and a reasonable coding rate of 1/2, so that covered users can enjoy a decent network access service. Indeed, we have evaluated in Figure 5.13(a) the impact of varying the SINR values studied in [136], and observed that higher values let coverage decay considerably. In contrast, lower values do not present much relevant impact on coverage, albeit they guarantee worse data rates. Moreover, the maximum allowed number of users served by a *gNB* or an *aBS* is set here to 100 users, which reflects the capacity scale of current 3GPP-compliant cells in terms of active users per BS sector.

In Figure E.1, we show the minimum data rate experienced by covered users. The rate is computed with the Shannon formula, using simulated SINR values for a network of 1000 UEs with up to 10 *aBS*s. With the selected SINR thresholds (10.9 dB) and maximum number of users per base station (100), and with a bandwidth of 20 MHz, the guaranteed data rate is 0.72 Mbps, which is plotted as an horizontal dotted line in the figure.

On the one side, since in the *PPP* scenario the distribution of users is uniform, the footprint of *gNB*s and *aBS*s is not sufficient to cover the maximum allowed number of users. Hence, resources are split among less users, so that they experience rates well above the guaranteed one. On the other side, scenarios like *Cheese* and, more clearly, *Capital* present a non-uniform distribution of users, with groups that form spontaneously. Hence, as previously seen in Figure 5.12, coverage is higher than for the *PPP* scenario. The

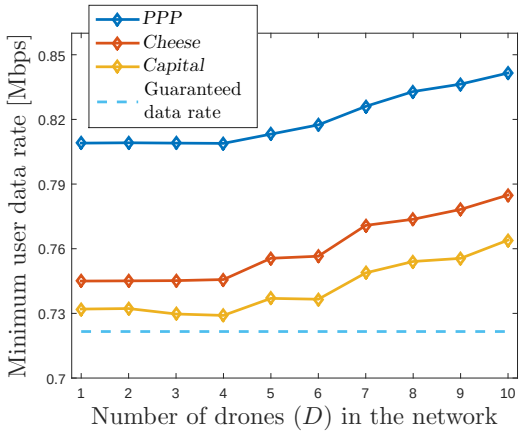


Figure E.1: Minimum user data rate achieved in comparison with the user guaranteed rate. $U = 1000$. Scenario: *dense*.

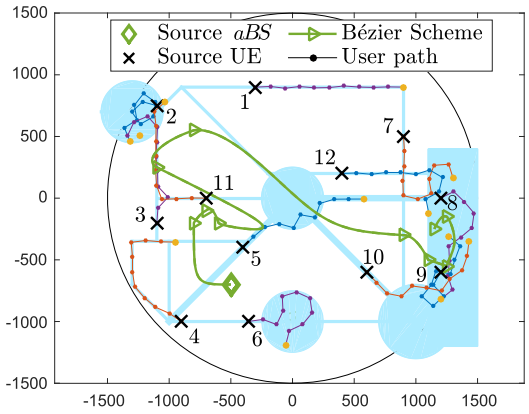


Figure E.2: Network dynamics in a small scenario during 10 minutes. $D = 1$, $U = 12$. Scenario: *high-rise, Capital*.

result is that both *gNB*s and *aBS*s tend to cover almost as many users as they are allowed to cover, and minimum rates approach the guaranteed one. In all scenarios, increasing the number of drones is beneficial because the resulting number of users per base station diminishes, on average.

The analysis of other scenarios, thresholds, limits in the number of users per base stations and in the backhaul, and environmental conditions lead to very similar qualitative conclusions, so we do not include those results in the figure.

Overall, here we have shown that the good performance discussed for our coverage optimization schemes, are not obtained in change of poor or, what would be worst, uncontrollably low data rates. In contrast, the guaranteed rate computed based on a fistful of optimization constraints in our problem formulation, results in a realistically close approximation for the lower bound of users performance.

E.2. User Mobility and Drone Trajectories

In order to evaluate the relation between user mobility and drone trajectories, in Figure E.2 we picture a small scenario with 12 UEs and 1 *aBS*. We depict both the *aBS* movement *and* the users movement, during 10 minutes. Here, to further simply presentation, we have took out *gNB*s.

In the figure, trajectories are marked with positions sampled once per minute, although the actual movement followed by users is more erratic, according to the random way-point model simulated with a one-second time resolution. An X marks the starting point for each user, while a green diamond indicates the initial position of the drone.

If we analyze the trajectories minute by minute, we can observe the evolution of

optimum coverage over time. For instance, at the beginning, the drone only covers users 4, 5 and 6, which is optimum at that stage (other users are far and there are no larger groups to cover). However, after one minute, these three users walk away from each other making not possible to cover them altogether, so that the *aBS* immediately reacts and flies where it can cover other three users. Note that, on its first-minute trajectory, the drone deflects its path towards user 4 to cover it momentarily. The effects caused by using the *Bézier Scheme* are evident throughout the entire trajectory of the drone. During the following minutes, the drone hovers on the left side of the map, until the users begin to gather in the area on the right part of the map. Hence, the drone quickly reacts to this change and gets repositioned on this region, where it stays for the last few minutes, eventually covering four users.

From what presented in this appendix, the proposed `OnDrone` heuristic, with the help of *Bézier* trajectories, is suitable for fast reconfiguration and therefore allows to dynamically adjust network coverage at the same time-scale at which the topology of users evolves. The presence of users plays the role of "attractor" for both optimizing the position of drones and for shaping the trajectories followed while repositioning.

F Optimal α-Fair Bandwidth Allocation in Wireless Backhaul Relay Networks

Bandwidth allocation is done by solving the convex optimization program presented in (6.4). We apply KKT conditions [156] so to find solutions without having to resort to a solver. To simplify the notation, we consider that, given a set of generic entities $\mathcal{E}$, there is a bijective function $\sigma : \{1, \ldots, |\mathcal{E}|\} \longrightarrow \mathcal{E}$ that maps entities onto integer numbers. With that, we can denote, e.g., the throughput of a user u indistinctly as T_u (with u being a user) or T_i (with i being an integer number), where $\sigma(i) = u$.

We next show how to solve the bandwidth allocation problem by considering two separate cases: with and without drones.

F.1. Without Drones

In this scenario, there are no drone. CP (6.4) simplifies considerably since *gNB* g only needs to manage resources to be split among *gNB*-served users, i.e., the set $\mathcal{U}_g$. Hence, we solve the following CP:

$$\begin{cases} \max\limits_{w_u, T_u} U^{\alpha}_{\text{cvx},g} = \begin{cases} \sum\limits_{u \in \mathcal{U}_g} (T_u)^{1-\alpha} \cdot \frac{1}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{u \in \mathcal{U}_g} \log\left(T_u\right), & \alpha = 1; \end{cases} \\ \text{s.t.:} \\ w_u \geq W^{\min}_{\mathcal{G}}, \qquad \forall u \in \mathcal{U}_g; \\ \sum\limits_{u \in \mathcal{U}_g} w_u = W_{\mathcal{G}}; \\ T_u \leq w_u \log_2(1+\gamma_{b,u}), \quad \forall u \in \mathcal{U}_g; \\ \sum\limits_{u \in \mathcal{U}_g} T_u \leq \tau_g. \end{cases} \tag{F.1}$$

Denote as $x = [\{w_u\}_{u \in \mathcal{U}_g}, \{T_u\}_{u \in \mathcal{U}_g}] \in \mathbb{R}^{2|\mathcal{U}_g|}$ the vector of variables. Then, depending on the value of α, we define the KKT functions as follows:

$$f^{\alpha}(x) = -\sum_{u\in\mathcal{U}_g} T_u^{1-\alpha} \qquad \text{if } \alpha \neq 1; \tag{F.2}$$

$$f^{1}(x) = -\sum_{u\in\mathcal{U}_g} \log T_u \qquad \text{if } \alpha = 1; \tag{F.3}$$

$$g_u(x) = W_{\mathcal{G}}^{\min} - w_u, \qquad \forall u \in \mathcal{U}_g; \tag{F.4}$$

$$g_{|\mathcal{U}_g|+u}(x) = T_u - w_u \log_2(1+\gamma_{g,u}), \quad \forall u \in \mathcal{U}_g; \tag{F.5}$$

$$g_{2|\mathcal{U}_g|+1}(x) = \sum_{u\in\mathcal{U}_g} T_u - \tau_g; \tag{F.6}$$

$$h(x) = \sum_{u\in\mathcal{U}_g} w_u - W_{\mathcal{G}}. \tag{F.7}$$

The corresponding KKT gradients are as follows:

$$\nabla f^{\alpha}(x) = \left[0, \ldots, \{(\alpha-1)T_u^{-\alpha}\}_{u\in\mathcal{U}_g}\right]; \tag{F.8}$$

$$\nabla f^{\log}(x) = \left[0, \ldots, 0, \left\{\tfrac{-1}{T_u}\right\}_{u\in\mathcal{U}_g}\right]; \tag{F.9}$$

$$\nabla g_u(x) = [0, \ldots, 0, -1|_u, 0, \ldots, 0], \quad \forall u\in\mathcal{U}_g; \tag{F.10}$$

$$\nabla g_{|\mathcal{U}_g|+u}(x) = \left[0, ..., 0, -\log_2(1+\gamma_{g,u})|_u, 0, ..., 0, 1|_{|\mathcal{U}_g|+u}, 0, ..., 0\right], \quad \forall u\in\mathcal{U}_g; \tag{F.11}$$

$$\nabla g_{2|\mathcal{U}_g|+1}(x) = \left[0, \ldots, 0, 1|_{|\mathcal{U}_g|+1}, 1, \ldots, 1\right]; \tag{F.12}$$

$$\nabla h(x) = \left[1, \ldots, 1|_{|\mathcal{U}_g|}, 0, \ldots, 0\right]. \tag{F.13}$$

Now, the KKT conditions state that if we find a vector $x^* \in \mathbb{R}^{2|\mathcal{U}_g|}$ that is feasible (i.e., it satisfy the original constraints of the problem), and multipliers $\mu_i \geq 0\ \forall 1 \leq i \leq 2|\mathcal{U}_g|+1$ and ν, and not all of them null, then the following equations hold and x^* is the optimal solution:

$$\overrightarrow{0} = \nabla f(x^*) + \sum_{i=1}^{2|\mathcal{U}_g|+1} \mu_i \nabla g_i(x^*) + \nu \nabla h(x^*); \tag{F.14}$$

$$\mu_i g_i(x^*) = 0, \qquad \forall 1 \leq i \leq 2|\mathcal{U}_g| + 1; \tag{F.15}$$

$$h(x^*) = 0, \tag{F.16}$$

where $f \in \{\{f^{\alpha}\}_{\alpha\in[0,1[}, f^{\log}\}$.

Hence, in what follows we describe how we have found the optimal solution for each particular case. We analyze separately the `MaxThr` case (when $\alpha = 0$), the α`Fair` case (when $\alpha \in]0, 1[$), the `PropFair` case (when $\alpha = 1$) and the `MaxMin` case (when $\alpha \to \infty$).

F.1.1. α`Fair` optimum ($\alpha \in]0, 1[$)

In this case, $f(x) = f^{\alpha}(x) = -\sum_{u\in\mathcal{U}_g} T_u^{1-\alpha}$, and the KKT conditions are as follows:

$$-\mu_i - \log_2(1+\gamma_{g,i})\cdot\mu_{|\mathcal{U}_g|+i}+\nu = 0, \quad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.17}$$

$$\mu_{|\mathcal{U}_g|+i} + \mu_{2|\mathcal{U}_g|+1} - (1-\alpha)T_i^{-\alpha} = 0, \quad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.18}$$

$$\mu_i \cdot \left(W_{\mathcal{G}}^{\min} - w_i\right) = 0, \quad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.19}$$

$$\mu_{|\mathcal{U}_g|+i}\cdot(T_i - w_i \log_2(1+\gamma_{g,i})) = 0, \quad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.20}$$

$$\mu_{2|\mathcal{U}_g|+1} \cdot \Big(\sum_{u\in\mathcal{U}_g} T_u - \tau_g \Big) = 0; \tag{F.21}$$

$$\sum_{u\in\mathcal{U}_g} w_u - W_{\mathcal{G}} = 0. \tag{F.22}$$

With unbounded backbone capacity τ_g. Assuming that $\tau_g \to +\infty$, then $\sum_{u\in\mathcal{U}_g} T_u < \tau_g$ for all feasible x. Hence, according to Eq. (F.21), $\mu_{2|\mathcal{U}_g|+1} = 0$. Therefore, for $1 \le i \le |\mathcal{U}_g|$, Eq. (F.18) becomes $\mu_{|\mathcal{U}_g|+i} = (1-\alpha)T_i^{-\alpha}$, and Eq. (F.17) becomes $\nu = \mu_i + (1-\alpha)T_i^{-\alpha}\cdot\log_2(1+\gamma_{g,i})$. As ν is a constant, we search for expressions of μ_i and T_i that make ν independent of index i. For convenience, we define set $\mathcal{J}$ as those indices i for which w_i is the minimum guaranteed value, i.e.:

$$\mathcal{J} = \left\{ i \mid w_i = W_{\mathcal{G}}^{\min} \right\}. \tag{F.23}$$

We start by assuming that $\mu_i = 0$, $\forall i \notin \mathcal{J}$, so that (F.19) is satisfied $\forall i \notin \mathcal{J}$. Given the SINR values, this assumption makes T_i become a function of ν only, $\forall i \notin \mathcal{J}$, i.e., $T_i = \left(\frac{1-\alpha}{\nu}\log_2(1+\gamma_{g,i})\right)^{\frac{1}{\alpha}}$, $\forall i \notin \mathcal{J}$. In addition, as τ_g is unbounded, we can assume that T_i must take the highest possible value, which is the Shannon capacity $w_i\log_2(1+\gamma_{g,i})$, $\forall 1 \le i \le |\mathcal{U}_g|$. Hence, $w_i = \left(\frac{1-\alpha}{\nu}\right)^{\frac{1}{\alpha}} \left(\log_2(1+\gamma_{g,i})\right)^{\frac{1-\alpha}{\alpha}}$ for all cases in which $w_i > W_{\mathcal{G}}^{\min}$ (i.e., $\forall i \notin \mathcal{J}$) and $w_i = W_{\mathcal{G}}^{\min}, \forall i \in \mathcal{J}$. Replacing the above expression for w_i in Eq. (F.22), we obtain an equation in which ν is the only unknown to be derived. We therefore have obtained the following solution:

Algorithm 9 Derivation of set $\mathcal{J}$ in α`Fair` (unbounded τ_g).

Input: $\log_2(1+\gamma_{g,i})$, $\forall 1 \le i \le |\mathcal{U}_g|$.

1: Initialize: $\mathcal{J} \leftarrow \emptyset$; $w_i \leftarrow \dfrac{W_{\mathcal{G}} - |\mathcal{J}| \cdot W_{\mathcal{G}}^{\min}}{\sum\limits_{j \notin \mathcal{J}} \left(\frac{\log_2\left(1+\gamma_{g,j}\right)}{\log_2\left(1+\gamma_{g,i}\right)} \right)^{\frac{1-\alpha}{\alpha}}}$, $\forall i \notin \mathcal{J}$.

2: **while** $\exists i_1 \notin \mathcal{J} \mid w_{i_1} < W_{\mathcal{G}}^{\min}$ **do**

3: $\quad i_0 \leftarrow \arg\max_{i \notin \mathcal{J}} \log_2(1+\gamma_{g,i})$.

4: $\quad \mathcal{J} \leftarrow \mathcal{J} \cup \{i_0\}$.

5: $\quad w_i \leftarrow \dfrac{W_{\mathcal{G}} - |\mathcal{J}| \cdot W_{\mathcal{G}}^{\min}}{\sum\limits_{j \notin \mathcal{J}} \left(\frac{\log_2\left(1+\gamma_{g,j}\right)}{\log_2\left(1+\gamma_{g,i}\right)} \right)^{\frac{1-\alpha}{\alpha}}}$, $\forall i \notin \mathcal{J}$.

6: **end while**

$$w_i = W_{\mathcal{G}}^{\min}, \qquad \forall i \in \mathcal{J}; \tag{F.24}$$

$$w_i = \frac{W_{\mathcal{G}} - |\mathcal{J}| \cdot W_{\mathcal{G}}^{\min}}{\sum\limits_{j \notin \mathcal{J}} \left(\frac{\log_2(1+\gamma_{g,j})}{\log_2(1+\gamma_{g,i})} \right)^{\frac{1-\alpha}{\alpha}}}, \qquad \forall i \notin \mathcal{J}; \tag{F.25}$$

$$T_i = w_i \log_2(1+\gamma_{g,i}) \qquad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.26}$$

$$\mu_i = \nu - \log_2(1+\gamma_{g,i})(1-\alpha) T_i^{-\alpha}, \qquad \forall i \in \mathcal{J}; \tag{F.27}$$

$$\mu_i = 0, \qquad \forall i \notin \mathcal{J}; \tag{F.28}$$

$$\mu_{|\mathcal{U}_g|+i} = (1-\alpha)\, T_i^{-\alpha}, \qquad \forall 1 \le i \le |\mathcal{U}_g|; \tag{F.29}$$

$$\mu_{2|\mathcal{U}_g|+1} = 0; \tag{F.30}$$

$$\nu = (1-\alpha) \left(\frac{\sum\limits_{j \notin \mathcal{J}} \left(\log_2(1+\gamma_{g,j}) \right)^{\frac{1-\alpha}{\alpha}}}{W_{\mathcal{G}} - |\mathcal{J}| \cdot W_{\mathcal{G}}^{\min}} \right)^{\alpha}. \tag{F.31}$$

The solution we have built accomplishes all KKT equations and works also for the case $\sum\limits_{u \in \mathcal{U}_g} T_u = \tau_g$. In particular, note in Eq. (F.27) that $\mu_i \ge 0$, $\forall i \in \mathcal{J}$, which can be seen by replacing the expressions for T_i, w_i and ν in the equation and consider that, $\forall i \in \mathcal{J}$, $W_{\mathcal{G}}^{\min} \ge \dfrac{W_{\mathcal{G}} - |\mathcal{J}| \cdot W_{\mathcal{G}}^{\min}}{\sum\limits_{j \notin \mathcal{J}} \left(\frac{\log_2\left(1+\gamma_{g,j}\right)}{\log_2\left(1+\gamma_{g,i}\right)} \right)^{\frac{1-\alpha}{\alpha}}}$.

A valid set $\mathcal{J}$ always exists and it is built as shown in Algorithm 9. The algorithm starts with an empty set and, while it is not true that $w_i \ge W_{\mathcal{G}}^{\min}$, $\forall i \notin \mathcal{J}$, keeps adding to $\mathcal{J}$ one user at a time, the one with the highest SINR, which is the user that consumes less resources. The algorithm stops for sure, eventually after moving all users, with a set $\mathcal{J}$ for which KKT and feasibility conditions are satisfied, therefore the built solution is optimal.

With limited backbone capacity τ_g. In order to find the optimal solution in the general case, we start from the solution for $\tau_g \to +\infty$. In case this procedure yields

Algorithm 10 Optimum for strictly concave increasing utility

Input: $\{x_i\}_{i=1}^n$, X.

1: **while** $X - \sum_{i=1}^{n} x_i < 0$ **do**
2: $X_M \leftarrow \{x_i \mid x_i = \max_{1\le j\le n} x_j\}$.
3: $x_{M_2} \leftarrow \arg\max\{x_i \mid x_i \notin X_M\}$ (assume $|x_{M_2}| = 1$).
4: $R \leftarrow -X + \sum_{i=1}^{n} x_i > 0$.
5: $R_j \leftarrow \frac{R}{|X_M|}$, $\forall 1 \le j \le n$ with $x_j \in X_M$.
6: $x_j \leftarrow x_j - \min\{R_j, x_j - x_{M_2}\}, \forall 1 \le j \le n$ with $x_j \in X_M$.
7: **end while**

$\sum_{u\in\mathcal{U}_g} T_u \le \tau_g$, we have the optimal solution described above. Otherwise, we have that $\sum_{u\in\mathcal{U}_g} T_u > \tau_g$, which is unfeasible. However, we can build a feasible solution by decreasing some values of T_i, motivated by the consideration that the utility function is strictly concave, so that the least utility reduction is obtained by reducing the highest throughput value, as shown in Algorithm 10 (for $n = |\mathcal{U}_g|$, $\{x_i\}_{i=1}^n = \{T_u\}_{u\in\mathcal{U}_g}$ and $X = \tau_g$).

Algorithm 10 is iterative, and it is based on the following principle: Because of concavity, if only one user has maximal throughput $T' = \max_i\{T_i\}$, and second best throughput is T'', any throughput reduction from T' to $T' - y \ge T''$ maximizes utility for a reduction of y in $\sum_i T_i$; if we have two users with maximal throughput, and we have to reduce the sum of throughputs by y, then strict concavity assures that $2\,(T' - y/2)^{1-\alpha} > (T' - (y/2 - \epsilon))^{1-\alpha} + (T' - (y/2 + \epsilon))^{1-\alpha}, \forall \epsilon \in]0; y/2], y/2 \in]0, T' - T'']$, so the best aggregate utility is obtained by decreasing the two highest throughput values both to $T' - y/2$. As it is easy to see, in case of tie between n users, the best choice consists in reducing their throughputs to $T' - y/n, \forall y/n \in]0, T' - T'']$.

F.1.2. `MaxThr` optimum ($\alpha = 0$)

In this case, $f(x) = f^0(x) = -\sum_{u\in\mathcal{U}_g} T_u$, and the KKT conditions are like for the case $\alpha \in]0, 1[$, except $\alpha = 0$.

With unbounded backbone capacity τ_g. First, we look for a solution assuming that τ_g is unbounded, i.e., assuming that $\sum_{u\in\mathcal{U}_g} T_u < \tau_g$ for all feasible x. Hence, according to Eq. (F.21), $\mu_{2|\mathcal{U}_g|+1} = 0$. Also, according to Eq. (F.18), $\mu_{|\mathcal{U}_g|+i} = 1$, $\forall 1 \le i \le |\mathcal{U}_g|$, and according to Eq. (F.20), $T_i = w_i \log_2(1 + \gamma_{g,i})$, $\forall 1 \le i \le |\mathcal{U}_g|$. Moreover, according to Eq. (F.17), $\nu - \mu_i = \log_2(1 + \gamma_{g,i})$, $\forall 1 \le i \le |\mathcal{U}_g|$.

Now, let $i_0 = \arg\max_{1\le i\le|\mathcal{U}_g|} \log_2(1 + \gamma_{g,i})$ and let $w_i = W_{\mathcal{G}}^{\min}$, $\forall 1 \le i \ne i_0 \le |\mathcal{U}_g|$. Hence, according to Eq. (F.22),

$$w_{i_0} = W_{\mathcal{G}} - \sum_{\substack{i=1\\ i\ne i_0}}^{|\mathcal{U}_g|} w_i = W_{\mathcal{G}} - \sum_{\substack{i=1\\ i\ne i_0}}^{|\mathcal{U}_g|} W_{\mathcal{G}}^{\min} = W_{\mathcal{G}} - (|\mathcal{U}_g| - 1)W_{\mathcal{G}}^{\min}.$$

Algorithm 11 Optimum `MaxThr` in Backhaul-free scenario

1: Initialize: $T_{i_0} = \min\left(\tau_g,\ w_{i_0}\log_2\left(1+\gamma_{g,i_0}\right)\right)$, $T_i = 0$, $\forall 1 \leq i \neq i_0 \leq |\mathcal{U}_g|$, $\overline{\mathcal{U}_g} = \mathcal{U}_g \setminus \{i_0\}$.
2: **while** $\sum_{i=1}^{|\mathcal{U}_g|} T_i < \tau_g$ **and** $\overline{\mathcal{U}_g} \neq \emptyset$ **do**
3: $\quad j_0 = \arg\max_{j\in\overline{\mathcal{U}_g}} \log_2\left(1+\gamma_{g,j}\right)$.
4: $\quad T_{j_0} = \min\left(\tau_g - \sum_{i\in\mathcal{U}_g} T_i, w_{j_0}\log_2\left(1+\gamma_{g,j_0}\right)\right)$.
5: $\quad \overline{\mathcal{U}_g} = \overline{\mathcal{U}_g} \setminus \{j_0\}$.
6: **end while**

Now, let $\mu_{i_0} = 0$. Hence, according to Eq. (F.17), $\nu = \log_2\left(1+\gamma_{g,i_0}\right)$ and $\mu_i = \log_2\left(1+\gamma_{g,i_0}\right) - \log_2\left(1+\gamma_{g,i}\right)$, $\forall 1 \leq i \leq |\mathcal{U}_g|$ (note that $\mu_i \geq 0\ \forall 1 \leq i \leq |\mathcal{U}_g|$ by definition of i_0).

Since the found solution x is feasible and accomplishes all KKT conditions, this is necessary the optimal solution.

With limited backbone capacity τ_g. We can observe that the optimal solution when $\sum_{u\in\mathcal{U}_g} T_u < \tau_g$ corresponds to assigning the maximum possible amount of resources to the users with highest SINR. If we now consider a bounded value for τ_g, we can build the solution following the same scheme to build Algorithm 11, which finds the optimal solution, as explained in what follows.

Let $i_0 = \arg\max_{1\leq i\leq|\mathcal{U}_g|} \log_2\left(1+\gamma_{g,i}\right)$. Let $w_i = W_{\mathcal{G}}^{\min}$, $\forall 1 \leq i \neq i_0 \leq |\mathcal{U}_g|$ and $w_{i_0} = W_{\mathcal{G}} - \sum_{\substack{i=1\\ i\neq i_0}} W_{\mathcal{G}}^{\min} = W_{\mathcal{G}} - (|\mathcal{U}_g|-1)W_{\mathcal{G}}^{\min}$. In the algorithm, we initially assign $T_{i_0} = \min\left(\tau_g, w_{i_0}\log_2(1+\gamma_{g,i_0})\right)$, $T_i = 0$, $\forall 1 \leq i \neq i_0 \leq |\mathcal{U}_g|$, and define the set $\overline{\mathcal{U}_g} = \mathcal{U}_g \setminus \{i_0\}$ (step 1). Next, while $\sum_{i=1}^{|\mathcal{U}_g|} T_i < \tau_g$ and $\overline{\mathcal{U}_g} \neq \emptyset$, we do the following three assignments: (i) $j_0 = \arg\max_{j\in\overline{\mathcal{U}_g}} \log_2\left(1+\gamma_{g,j}\right)$ (step 3), (ii) $T_{j_0} = \min\left(\tau_g - \sum_{i\in\mathcal{U}_g} T_i, w_{j_0}\log_2\left(1+\gamma_{g,j_0}\right)\right)$ (step 4), and (iii) $\overline{\mathcal{U}_g} = \overline{\mathcal{U}_g} \setminus \{j_0\}$ (step 5). This procedure finds values of $\{T_i\}_{i=1}^{|\mathcal{U}_g|}$ such that $\sum_{i=1}^{|\mathcal{U}_g|} T_i \leq \tau_g$. Besides, if $\overline{\mathcal{U}_g} \neq \emptyset$ when the algorithm concludes, we have that $\sum_{i=1}^{|\mathcal{U}_g|} T_i = \tau_g$.

Note that in case $\sum_{i=1}^{|\mathcal{U}_g|} T_i < \tau_g$, the solution of Algorithm 11 corresponds with the solution for unbounded τ_g. Moreover, in case the output of the algorithm is such that $\sum_{i=1}^{|\mathcal{U}_g|} T_i = \tau_g$, the solution is also optimal because, with $\alpha = 0$, $\sum_{i=1}^{|\mathcal{U}_g|} T_i$ is the utility function, and τ_g is the maximum that that sum can achieve, according to the last constraint in (F.1). Therefore, the solution computed with Algorithm 11 is always optimal for $\alpha = 0$.

F.1.3. `PropFair` optimum ($\alpha = 1$)

This case aims to maximize the network proportional fairness metric, $f(x) = f^{\log}(x) = -\sum_{u \in \mathcal{U}_g} \log(T_u)$. The KKT conditions for this case are like for the case $\alpha \in]0, 1[$, except condition (F.18) is replaced by the following one:

$$\mu_{|\mathcal{U}_g|+i} + \mu_{2|\mathcal{U}_g|+1} - \frac{1}{T_i} = 0, \quad \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.32}$$

With unbounded backbone capacity τ_g. If τ_g is unbounded, Eq. (F.21) requires that $\mu_{2|\mathcal{U}_g|+1} = 0$. Hence, according to Eq. (F.32), $\mu_{|\mathcal{U}_g|+i} = \frac{1}{T_i}$, $\forall 1 \leq i \leq |\mathcal{U}_g|$.

Now, $\forall 1 \leq i \leq |\mathcal{U}_g|$, let $\mu_i = 0$ as in the previous cases, and $w_i = \frac{W_{\mathcal{G}}}{|\mathcal{U}_g|}$, $T_i = \frac{W_{\mathcal{G}}}{|\mathcal{U}_g|} \log_2(1 + \gamma_{g,i})$. Hence, according to Eq. (F.17), $\nu = \frac{|\mathcal{U}_g|}{W_{\mathcal{G}}}$.

Since the found feasible solution x accomplishes all KKT conditions, this is the optimal solution. Note that, as expected, in a `PropFair` without backbone constraints, all users would obtain the same amount of resources.

With limited backbone capacity τ_g. To find the optimal solution for the generic case, in which τ_g is limited, we start from the solution for unbounded backbone capacity. If this results in $\sum_{u \in \mathcal{U}_g} T_u > \tau_g$, being $\log(x)$ is a strictly concave increasing function, we can apply Algorithm 10 with $n = |\mathcal{U}_g|$, $\{x_i\}_{i=1}^n = \{T_u\}_{u \in \mathcal{U}_g}$ and $X = \tau_g$ in order to find the optimal solution of the `PropFair` case.

F.1.4. `MaxMin` optimum ($\alpha \to \infty$)

The utility function $f(x) = \min_{u \in \mathcal{U}_g} T_u$ is not differentiable, so we cannot directly apply KKT conditions. However, following the spirit of `MaxMin` optimization, we can reformulate the problem by adding a new decision variable T, changing the utility function and adding one extra set of constraints. This results in the following convex program:

$$\begin{cases} \max T; \\ \text{s.t.:} \\ T_u \geq T, & \forall u \in \mathcal{U}_g; \\ w_u \geq W_{\mathcal{G}}^{\min}, & \forall u \in \mathcal{U}_g; \\ \sum_{u \in \mathcal{U}_g} w_u = W_{\mathcal{G}}; \\ T_u \leq w_u \log_2(1 + \gamma_{b,u}), & \forall u \in \mathcal{U}_g; \\ \sum_{u \in \mathcal{U}_g} T_u \leq \tau_g. \end{cases} \tag{F.33}$$

Since we have added a new decision variable T, we denote the solution as $x = [\{w_u\}_{u \in \mathcal{U}_g}, \{T_u\}_{u \in \mathcal{U}_g}, T] \in \mathbb{R}^{2|\mathcal{U}_g|+1}$. Hence, we add to the KKT functions of Eqs. (F.2)–

(F.7) the following KKT functions:

$$f^{\min}(x) = -T; \tag{F.34}$$

$$g_{2|\mathcal{U}_g|+1+u}(x) = T - T_u, \quad \forall u \in \mathcal{U}_g. \tag{F.35}$$

Also, we have to add to KKT gradients of Eqs. (F.8)–(F.13) the following KKT gradients:

$$\nabla f^{\min}(x) = [0, \ldots, 0, -1]; \tag{F.36}$$

$$\nabla g_{2|\mathcal{U}_g|+1+u}(x) = [0, ..., 0, -1|_{|\mathcal{U}_g|+u}, 0, ..., 0, 1], \ \forall u \in \mathcal{U}_g. \tag{F.37}$$

In this case, applying KKT conditions, i.e., finding $\nu \in \mathbb{R}$ and positive constants $\{\mu_i\}_{i=1}^{3|\mathcal{U}_g|+1}$ not all of them null, and $x \in \mathbb{R}^{2|\mathcal{U}_g|+1}$ such that $\overrightarrow{0} = \nabla f^{\min}(x) + \sum_{i=1}^{3|\mathcal{U}_g|+1} \mu_i \nabla g_i(x) + \nu \nabla h(x)$ results into the following set of KKT equations:

$$-\mu_i - \log_2(1+\gamma_{g,i}) \cdot \mu_{|\mathcal{U}_g|+i} + \nu = 0, \ \ \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.38}$$

$$-\mu_{|\mathcal{U}_g|+i} + \mu_{2|\mathcal{U}_g|+1} - \mu_{2|\mathcal{U}_g|+1+i} = 0, \ \ \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.39}$$

$$-1 + \sum_{i=1}^{|\mathcal{U}_g|} \mu_{2|\mathcal{U}_g|+1+i} = 0; \tag{F.40}$$

$$\mu_i \cdot \left(W_{\mathcal{G}}^{\min} - w_i \right) = 0, \ \ \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.41}$$

$$\mu_{|\mathcal{U}_g|+i} \cdot (T_i - w_i \log_2(1+\gamma_{g,i})) = 0, \ \ \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.42}$$

$$\mu_{2|\mathcal{U}_g|+1} \cdot \Big(\sum_{u \in \mathcal{U}_g} T_u - \tau_g \Big) = 0; \tag{F.43}$$

$$\mu_{2|\mathcal{U}_g|+1+i} \cdot (T - T_i) = 0, \ \ \forall 1 \leq i \leq |\mathcal{U}_g|; \tag{F.44}$$

$$\sum_{u \in \mathcal{U}_g} w_u - W_{\mathcal{G}} = 0. \tag{F.45}$$

With unbounded backbone capacity τ_g. First, we look for a solution assuming that τ_g is unbounded. Hence, according to Eq. (F.43), $\mu_{2|\mathcal{U}_g|+1} = 0$. As $\{T_i\}_{i=1}^{|\mathcal{U}_g|}$ is not limited by τ_g, we take $T_i = w_i \log_2(1 + \gamma_{g,i})$, $\forall 1 \leq i \leq |\mathcal{U}_g|$.

We initially assume that $w_i = W_{\mathcal{G}}^{\min}$, $\forall 1 \leq i \leq |\mathcal{U}_g|$ and build an algorithm that finds the optimal solution of CP (F.33) (see Algorithm 12). Please note that now $T_i = W_{\mathcal{G}}^{\min} \cdot \log_2(1 + \gamma_{g,i})$, $\forall 1 \leq i \leq |\mathcal{U}_g|$ (step 1). We assume, without loss of generality, that $T_{i-1} \leq T_i$, $\forall 2 \leq i \leq |\mathcal{U}_g|$ (step 2), and define the set of indices $\mathcal{J}$ as those indices i such that T_i is minimum (step 3), i.e.,

$$\mathcal{J} = \left\{ i \in \{1, \ldots, |\mathcal{U}_g|\} \ | \ T_i = \min_{1 \leq j \leq |\mathcal{U}_g|} T_j \right\}. \tag{F.46}$$

Algorithm 12 works in the following way: while there are still bandwidth resources

Algorithm 12 Optimal solution of CP (F.33) (unbounded τ_g).

1: Initialize: $w_i \leftarrow W_{\mathcal{G}}^{\min}$, $T_i \leftarrow w_i \log_2(1+\gamma_{g,i})$, $\forall 1 \leq i \leq |\mathcal{U}_g|$.
2: Assume, WLOG, $T_{i-1} \leq T_i$, $\forall 2 \leq i \leq |\mathcal{U}_g|$.
3: $\mathcal{J} \leftarrow \left\{ i \in \{1, \ldots, |\mathcal{U}_g|\} \mid T_i = \min_{1 \leq j \leq |\mathcal{U}_g|} T_j \right\}$.
4: **while** $\sum_{i=1}^{|\mathcal{U}_g|} w_i < W_{\mathcal{G}}$ **and** $|\mathcal{J}| \neq |\mathcal{U}_g|$ **do**
5: $j_0 \leftarrow \arg\min\{T_i \mid i \notin \mathcal{J}\}$ **and** $K_b \leftarrow 1$.
6: $k_i \leftarrow \frac{T_{j_0}}{\log_2(1+\gamma_{g,i})} - w_i$, $\forall i \in \mathcal{J}$.
7: **if** $\sum_{i \in \mathcal{J}} k_i > W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i$ **then**
8: $k_i \leftarrow \frac{W_{\mathcal{G}} - \sum_{i \notin \mathcal{J}} w_i}{\log_2(1+\gamma_{g,i}) \sum_{i \in \mathcal{J}} \frac{1}{\log_2\left(1+\gamma_{g,i}\right)}} - w_i$, $\forall i \in \mathcal{J}$ **and** $K_b \leftarrow 0$.
9: **end if**
10: $w_i \leftarrow w_i + k_i$, $\forall i \in \mathcal{J}$ **and** $T_i \leftarrow w_i \log_2(1+\gamma_{g,i})$, $\forall i \in \mathcal{J}$.
11: **if** $K_b = 1$ **then**
12: $\mathcal{J} \leftarrow \mathcal{J} \cup \{j_0\}$.
13: **end if**
14: **end while**
15: $T \leftarrow \min_{1 \leq i \leq |\mathcal{U}_g|} T_i$.

to be allocated, i.e., $\sum_{i=1}^{|\mathcal{U}_g|} w_i < W_{\mathcal{G}}$, and $|\mathcal{J}| \neq |\mathcal{U}_g|$, we select the index $j_0 = \arg\min\{T_i \mid i \notin \mathcal{J}\}$ such that T_{j_0} is the lowest user throughput rate not equal to the minimum of the throughput rates (step 5). Now, we aim to increase w_i, $\forall i \in \mathcal{J}$ as much as possible in a way that is max–min fair and $T_i \leq T_{j_0}$, $\forall i \in \mathcal{J}$. Hence, we have to find $\{k_i\}_{i \in \mathcal{J}}$ such that $\{w_i\}_{i \in \mathcal{J}}$ are increased by k_i each. The optimal way of doing this is by first assigning $k_i = \frac{T_{j_0}}{\log_2(1+\gamma_{g,i})} - w_i$, $\forall i \in \mathcal{J}$ (step 6) and checking that $\sum_{i \in \mathcal{J}} k_i \leq W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i$. In case such an inequality is not satisfied, then assign $k_i = \frac{W_{\mathcal{G}} - \sum_{i \notin \mathcal{J}} w_i}{\log_2(1+\gamma_{g,i}) \sum_{i \in \mathcal{J}} \frac{1}{\log_2\left(1+\gamma_{g,i}\right)}} - w_i$, $\forall i \in \mathcal{J}$ (step 8).

Once k_i is derived, we assign $w_i \leftarrow w_i + k_i$, $\forall i \in \mathcal{J}$ (step 10). Now, in case that we have assigned $k_i = \frac{T_{j_0}}{\log_2(1+\gamma_{g,i})} - w_i$, $\forall i \in \mathcal{J}$, we reassign set $\mathcal{J}$ as $\mathcal{J} \leftarrow \mathcal{J} \cup \{j_0\}$ (step 12), and start all over. Otherwise, the algorithm stops and the optimal solution is found.

In Note 1 we prove that the assigned $\{k_i\}_{i \in \mathcal{J}}$ at each iteration of Algorithm 12 provides the optimal max–min fair distribution of resources.

Note 1. *Given a distribution of resources* $\{w_i\}_{i=1}^{|\mathcal{U}_g|}$ *and users throughput rates* $\{T_i\}_{i=1}^{|\mathcal{U}_g|}$ *such that* $T_i = w_i \log_2(1+\gamma_{g,i})$, $\forall 1 \leq i \leq |\mathcal{U}_g|$, *we define the set* $\mathcal{J}$ *as in Eq.* (F.46).

Hence, we have that $w_i \log_2(1+\gamma_{g,i}) = w_k \log_2(1+\gamma_{g,k})$, $\forall i, k \in \mathcal{J}$.

Now, given $j_0 = \arg\min\{T_i \mid i \notin \mathcal{J}\}$, *we want to increase* $\{w_i\}_{i \in \mathcal{J}}$ *as much as*

possible by k_i each in a max–min *fair way so that* $(w_i + k_i)\log_2(1+\gamma_{g,i}) \leq T_{j_0}$, $\forall i \in \mathcal{J}$. *Hence, we must solve the following convex program:*

$$\begin{cases} \max \min\limits_{i\in\mathcal{J}} (w_i + k_i)\log_2(1+\gamma_{g,i}); \\ s.t.: \\ (w_i + k_i)\log_2(1+\gamma_{g,i}) \leq T_{j_0}, \qquad \forall i \in \mathcal{J}; \\ \sum\limits_{i=1}^{|\mathcal{U}_g|} w_i + \sum\limits_{i\in\mathcal{J}} k_i \leq W_{\mathcal{G}}; \end{cases} \tag{F.47}$$

Since CP (F.47) *has a* max–min *utility function, we reformulate this CP into an equivalent CP to which we can directly apply KKT conditions:*

$$\begin{cases} \max L; \\ s.t.: \\ (w_i + k_i)\log_2(1+\gamma_{g,i}) \geq L, \quad \forall i \in \mathcal{J}; \\ k_i \leq \frac{T_{j_0}}{\log_2(1+\gamma_{g,i})} - w_i, \qquad \forall i \in \mathcal{J}; \\ \sum\limits_{i\in\mathcal{J}} k_i \leq W_{\mathcal{G}} - \sum\limits_{i=1}^{|\mathcal{U}_g|} w_i; \end{cases} \tag{F.48}$$

Please, note that the first constraint of CP (F.48) *can be rearranged as* $k_i \geq \frac{L}{\log_2(1+\gamma_{g,i})} - w_i$, $\forall i \in \mathcal{J}$. *Also, the second and third constraints of CP* (F.48) *are equivalent to the constraints of the* max–min *CP* (F.47).

Now, we make use of KKT conditions to solve CP (F.48). *The decision variables are gathered in vector* $x = [\{k_i\}_{i\in\mathcal{J}}, L] \in \mathbb{R}^{|\mathcal{J}|+1}$. *We define the following KKT functions:*

$$f(x) = -L; \tag{F.49}$$

$$g_i(x) = \frac{L}{\log_2(1+\gamma_{g,i})} - k_i - w_i, \quad \forall i \in \mathcal{J}; \tag{F.50}$$

$$g_{|\mathcal{J}|+i}(x) = k_i - \frac{T_{j_0}}{\log_2(1+\gamma_{g,i})} - w_i, \quad \forall i \in \mathcal{J}; \tag{F.51}$$

$$g_{2|\mathcal{J}|+1}(x) = \sum_{i\in\mathcal{J}} k_i - W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i, \quad \forall i \in \mathcal{J}. \tag{F.52}$$

Hence, the KKT gradients are the following:

$$\nabla f(x) = [0, \ldots, 0, 1]; \tag{F.53}$$

$$\nabla g_i(x) = \left[0, ..., 0, -1|_i, 0, ..., 0, \tfrac{1}{\log_2(1+\gamma_{g,i})}\right], \ \forall i \in \mathcal{J}; \tag{F.54}$$

$$\nabla g_{|\mathcal{J}|+i}(x) = [0, \ldots, 0, 1|_i, 0, \ldots, 0], \qquad \forall i \in \mathcal{J}; \tag{F.55}$$

$$\nabla g_{2|\mathcal{J}|+1}(x) = [1, \ldots, 1, 0]. \tag{F.56}$$

We derive the following KKT conditions:

$$-\mu_i + \mu_{|\mathcal{J}|+i} + \mu_{2|\mathcal{J}|+1} = 0, \quad \forall i \in \mathcal{J}; \tag{F.57}$$

$$-1 + \sum_{i=1}^{|\mathcal{J}|} \frac{\mu_i}{\log_2(1+\gamma_{g,i})} = 0; \tag{F.58}$$

$$\mu_i \cdot \left(\frac{L}{\log_2(1+\gamma_{g,i})} - k_i - w_i \right) = 0, \quad \forall i \in \mathcal{J}; \tag{F.59}$$

$$\mu_{|\mathcal{J}|+i} \cdot \left(k_i - \frac{T_{j0}}{\log_2(1+\gamma_{g,i})} + w_i \right) = 0, \quad \forall i \in \mathcal{J}; \tag{F.60}$$

$$\mu_{2|\mathcal{J}|+1} \cdot \left(\sum_{i\in\mathcal{J}} k_i - W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i \right) = 0. \tag{F.61}$$

First, we note that if assigning $k_i = \frac{T_{j0}}{\log_2(1+\gamma_{g,i})} - w_i$, $\forall i \in \mathcal{J}$ *accomplishes that* $\sum_{i\in\mathcal{J}} k_i \leq W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i$, *we have the optimal solution, as each* k_i *receives the maximum possible value and constraints are satisfied.*

Hence, we can assume that $\exists i_0 \in \mathcal{J} \mid k_{i_0} < \frac{T_{j0}}{\log_2\left(1+\gamma_{g,i_0}\right)} - w_{i_0}$ *and hence, according to Eq.* (F.60), $\mu_{|\mathcal{J}|+i_0} = 0$.

The optimal $\{k_i\}_{i\in\mathcal{J}}$ *and* L *that solve the KKT conditions are:*

$$k_i = \frac{W_{\mathcal{G}} - \sum_{i\notin\mathcal{J}} w_i}{\log_2(1+\gamma_{g,i}) \sum_{i\in\mathcal{J}} \frac{1}{\log_2\left(1+\gamma_{g,i}\right)}} - w_i, \quad \forall i \in \mathcal{J}; \tag{F.62}$$

$$L = \frac{W_{\mathcal{G}} - \sum_{i\notin\mathcal{J}} w_i}{\sum_{i\in\mathcal{J}} \frac{1}{\log_2\left(1+\gamma_{g,i}\right)}}. \tag{F.63}$$

It is simple to check that $\sum_{i\in\mathcal{J}} k_i = W_{\mathcal{G}} - \sum_{i=1}^{|\mathcal{U}_g|} w_i$, *so that Eq.* (F.61) *is satisfied.*

Now, we assign $\mu_i = \frac{w_i}{\sum_{i\in\mathcal{J}} w_i} \log_2(1+\gamma_{g,i})$, $\forall i \in \mathcal{J}$, *and hence, since* $\mu_{|\mathcal{J}|+i_0} = 0$, *and according to Eq.* (F.57), *we have that* $\mu_{2|\mathcal{J}|+1} = \mu_{i_0} = \frac{w_{i_0}}{\sum_{i\in\mathcal{J}} w_i} \log_2(1+\gamma_{g,i_0})$. *Hence, according also to Eq.* (F.57), $\mu_{|\mathcal{J}|+i} = \mu_i - \mu_{2|\mathcal{J}|+1} = \frac{w_i}{\sum_{i\in\mathcal{J}} w_i} \log_2(1+\gamma_{g,i}) - \frac{w_{i_0}}{\sum_{i\in\mathcal{J}} w_i} \log_2(1+\gamma_{g,i_0})$, $\forall i \in \mathcal{J}$. *Since* $i, i_0 \in \mathcal{J}$, *we have that, by definition on* $\mathcal{J}$, $\mu_{|\mathcal{J}|+i} = 0$, $\forall i \in \mathcal{J}$.

As a result, since with such a solution all KKT conditions, are satisfied, we have the optimal solution to CP (F.48).

With limited backbone capacity τ_g. Now, we assume that τ_g is bounded. In order to find the optimal solution under this assumption, we first solve the problem assuming

unbounded τ_g, as detailed above. If such an output provides a feasible solution, i.e., $\sum_{u \in \mathcal{U}_g} T_u \leq \tau_g$, we have the optimal solution. Hence, we assume that such an output provides an unfeasible solution, i.e., $\sum_{u \in \mathcal{U}_g} T_u > \tau_g$.

Let $\{w_u\}_{u \in \mathcal{U}_g}$, $\{T_u\}_{u \in \mathcal{U}_g}$ be the unfeasible solution provided by the max–min optimization with unbounded τ_g, and let $T = \min_{u \in \mathcal{U}_g} \{T_u\}$ be the minimum achieved throughput by users so far (note that T is the value of the utility). Due to the max–min fairness nature, every user u such that $T_u > T$ disposes of the minimum amount of resources, $W_{\mathcal{G}}^{\min}$ and $T_u = W_{\mathcal{G}}^{\min} \log_2(1 + \gamma_{g,u})$, $\forall u \in \mathcal{U}_g \mid T_u > T$ (otherwise, if such users disposed of more than $W_{\mathcal{G}}^{\min}$ resources, such exceeded resources could be reallocated to those users with minimum throughput to increase the utility function, which is not possible from the output max–min fairness optimization with unbounded τ_g).

Hence, as no resources can be removed from any user u such that $T_u > T$, and $\sum_{u \in \mathcal{U}_g} T_u > \tau_g$, we can reduce the rate of these users to, for instance, $T_u = T$ (hence, now $T_u = T$, $\forall u \in \mathcal{U}_g$). If now $\sum_{u \in \mathcal{U}_g} T_u \leq \tau_g$, we have found the optimal solution. Otherwise, we need to reduce more individual throughput rates. In order to be max–min fair, we have to reduce every individual rate the same amount until the τ_g–constraint is satisfied. Hence, we define $R = \sum_{u \in \mathcal{U}_g} T_u - \tau_g > 0$ and decrease every individual rate by $\frac{R}{|\mathcal{U}_g|}$, i.e., $T_u = T - \frac{R}{|\mathcal{U}_g|}$, $\forall u \in \mathcal{U}_g$. Hence, the max–min fairness optimization of CP (F.33) is solved also under the assumption of bounded τ_g.

F.2. With One Drone

In this scenario, we consider that *gNB* g serves its users $\mathcal{U}_g$ and also one *aBS* a, i.e., $\exists a \in \mathcal{A} \mid \mathcal{A}_g = \{a\}$. Hence, CP (6.4) simplifies since backhaul resources do not need to be split over multiple *aBS*s, but all of them are assigned to only one *aBS*, i.e., $w^a = W_{\mathcal{B}}$. Hence, we solve the following CP:

$$
\begin{cases}
\max\limits_{w_u, T_u} U^{\alpha}_{\text{cvx},g} = \begin{cases} \sum\limits_{u \in \mathcal{U}_g \cup \mathcal{U}_a} (T_u)^{1-\alpha} \cdot \frac{1}{1-\alpha}, & \alpha \neq 1; \\ \sum\limits_{u \in \mathcal{U}_g \cup \mathcal{U}_a} \log\left(T_u\right), & \alpha = 1; \end{cases} \\
\text{s.t.:} \\
T^a \leq W_{\mathcal{B}} \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right); \\
w_u \geq W^{\min}_{\mathcal{G}}, \qquad \forall u \in \mathcal{U}_g; \\
\sum\limits_{u \in \mathcal{U}_g} w_u = W_{\mathcal{G}}; \\
w_u \geq W^{\min}_{\mathcal{A}}, \qquad \forall u \in \mathcal{U}_a; \\
\sum\limits_{u \in \mathcal{U}_a} w_u = W_{\mathcal{A}}; \\
T_u \leq w_u \log_2(1+\gamma_{g,u}), \qquad \forall u \in \mathcal{U}_g; \\
T_u \leq w_u \log_2(1+\gamma_{a,u}), \qquad \forall u \in \mathcal{U}_a; \\
\sum\limits_{u \in \mathcal{U}_a} T_u \leq T^a; \\
\sum\limits_{u \in \mathcal{U}_g} T_u + T^a \leq \tau_g.
\end{cases}
\tag{F.64}
$$

In order to solve CP (F.64), we do not derive KKT conditions, as we will make use of the analysis conducted in Subection F.1.

With unbounded backbone capacity τ_g. First, we assume that τ_g is unbounded, and hence users throughput rates are only limited by their Shannon capacity, according to the allocated bandwidth resources. As *gNB* g disposes of two independent baskets of resources, $W_{\mathcal{G}}$ for *gNB*-served users, and $W_{\mathcal{B}}$ for the backhaul-served *aBS*, the distribution of *gNB*-served users resources disregards from *aBS*-served users resources (since τ_g is unbounded). Hence, the backhaul rate for *aBS* a, T^a, is limited only by the Shannon capacity, so that we can assume that $T^a = W_{\mathcal{B}} \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)$.

Hence, we solve optimal resource allocation for *gNB*-served users assuming unbounded τ_g, and solve also optimal resource allocation for *aBS*-served users assuming that their aggregated throughput is limited by the backhaul capacity T^a, i.e., $\sum\limits_{u \in \mathcal{U}_a} T_u \leq T^a$. Both cases can be solved as detailed in Subsection F.1. As a result, CP (F.64) is solved under the assumption of unbounded τ_g, for any value of α, including $\alpha \to \infty$.

The solution provided is optimal although not necessarily unique, as the backhaul capacity T^a might be higher than the aggregated served throughputs. Hence, to ease the understanding of upcoming sections, we assume that the provided optimal solution satisfies that $T^a = \sum\limits_{u \in \mathcal{U}_a} T_u$, which remains feasible and optimal.

With limited backbone capacity τ_g. Now, we do not assume that τ_g is unbounded. In order to solve this case, we first search the optimal solution assuming that τ_g is indeed unbounded and checking if $\sum\limits_{u \in \mathcal{U}_g} T_u + T^a = \sum\limits_{u \in \mathcal{U}_g} T_u + \sum\limits_{u \in \mathcal{U}_a} T_u \leq \tau_g$. In case the inequality

is satisfied, we have found the optimal solution. Otherwise, users throughputs must be decreased in order to satisfy the τ_g–constraint.

Hence, according to what proven and described in Subsection F.1, we can decrease the convenient users throughputs T_u until we get to satisfy that $\sum_{u\in\mathcal{U}_g} T_u + T^a = \sum_{u\in\mathcal{U}_g} T_u + \sum_{u\in\mathcal{U}_a} T_u = \tau_g$ and hence provide the optimal solution.

F.3. Generic Case

In this scenario, we consider the generic case, formulated in CP (6.4). As in the previous cases, we distinguish between unbounded and bounded τ_g assumptions. In Algorithm 13 we show how to find the optimal solution, as detailed in what follows.

With unbounded backbone capacity τ_g. Here, we assume that τ_g is unbounded, so that distribution of user resources at *gNB*s is not affected by the presence of *aBS*s and their associated users. However, now we cannot know a priori how many resources each *aBS* will get, as they share a common bandwidth of $W_{\mathcal{B}}$. Each *aBS* must receive w^a resources according to the optimization output. We find the optimal solution as detailed as follows.

First, the optimal resource allocation for *gNB*-served users is as previously detailed in Section F.1. Then, for each *aBS* $a \in \mathcal{A}_g$ we solve also optimal resource allocation assuming unbounded backhaul capacity T^a, i.e., *aBS*-served users throughput rates are limited only by their Shannon capacity, according to the bandwidth allocated, using the scheme detailed in Section F.1. For convenience, we now denote by $T_{a,u}$ the throughput of access link (a,u), $\forall a \in \mathcal{A}_g\ \forall u \in \mathcal{U}_a$. For each $a \in \mathcal{A}_g$, we need a backhaul capacity of $T^a = \sum_{u\in\mathcal{U}_a} T_{a,u}$. Hence, *aBS* a needs $w^a = \max\left(W_{\mathcal{B}}^{\min}, \sum_{u\in\mathcal{U}_a} \frac{T_{a,u}}{\log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)}\right)$.

Now, in case $\sum_{a\in\mathcal{A}_g} w^a \leq W_{\mathcal{B}}$, we choose $a_0 \in \mathcal{A}_g$ arbitrarily and add resources to w^{a_0} so that we get to $\sum_{a\in\mathcal{A}_g} w^a = W_{\mathcal{B}}$ (i.e., $w^{a_0} = w^{a_0} + W_{\mathcal{B}} - \sum_{a\in\mathcal{A}_g} w^a$). Hence, in this case the optimization is solved.

Conversely, in case $\sum_{a\in\mathcal{A}_g} w^a > W_{\mathcal{B}}$, we are assigning to backhaul *aBS*s more resources than what available at the *gNB*. Hence, some resources should be removed. In this case, we need to build the optimal solution from scratch. First, we assign to each *aBS* $a \in \mathcal{A}_g$ the minimum bandwidth: $w^a = W_{\mathcal{B}}^{\min}$, $\forall a \in \mathcal{A}_g$ (step 1 in Algorithm 13). Second, we assign to each *aBS* $a \in \mathcal{A}_g$ the highest achievable throughput: $T^a = w^a \log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)$, $\forall a \in \mathcal{A}_g$ (step 2). Third, we solve optimal resource allocation at each $a \in \mathcal{A}_g$ and also at *gNB* g. Hence, we dispose of $\{T_{a,u}\}_{u\in\mathcal{U}_a}$, $\forall a \in \mathcal{A}_g$, and of $\{T_u\}_{u\in\mathcal{U}_g}$ (step 3). Please note that $\forall a \in \mathcal{A}_g$, $\forall u \in \mathcal{U}_a$ it might happen that $T_{a,u} < w_u \log_2(1+\gamma_{a,u})$ because of the backhaul limitation T^a. Hence, if T^a increases, such $T_{a,u}$'s can also increase. The contribution of $\{T_{a,u}\}_{u\in\mathcal{U}_a}$, $\forall a \in \mathcal{A}_g$ to utility $\mathcal{U}_{\text{cvx},g}^{\alpha}$ is either $(T_{a,u})^{1-\alpha}$ or $\log(T_{a,u})$. Hence, the

distribution of throughput contribution follows an increasing and concave function. Since $\xi(x) = x^{1-\alpha}$ and $\xi(x) = \log(x)$ are both increasing and concave functions, the way of increasing most utility $\mathcal{U}^{\alpha}_{\text{cvx},g}$ is by raising the lowest values of $\{T_{a,u}\}_{u\in\mathcal{U}_a}$, $\forall a \in \mathcal{A}_g$ as much as possible as long as constraints are not violated. Also in the max–min case: the way of increasing as much as possible the utility is by equally raising the lowest values of $\{T_{a,u}\}_{u\in\mathcal{U}_a}$, $\forall a \in \mathcal{A}_g$, as long as constraints are not violated, due to the well-known max–min fairness nature. Let

$$T_m = \min\{T_{a,u} \mid T_{a,u} < w_u \log_2(1+\gamma_{a,u}), a \in \mathcal{A}_g, u \in \mathcal{U}_a\} \tag{F.65}$$

(step 6) be the minimum throughput rate that has not reached the corresponding to the Shannon capacity (note that if such a T_m does not exist, we are done). Let

$$\mathcal{L}_m = \left\{(a,u) \in \mathcal{A}_g \times \bigcup_{a\in\mathcal{A}_g} \mathcal{U}_a \mid T_{a,u} = T_m\right\} \tag{F.66}$$

be the set of those *aBS*–UE links such that the link rate is the same as the minimum T_m (step 7). Let

$$T_M = \min\{T_{a,u} \mid T_{a,u} > T_m, a \in \mathcal{A}_g, u \in \mathcal{U}_a\} \tag{F.67}$$

be the minimum throughput rate among those rates that are not as the minimum T_m (step 8). Let

$$T_{M_2} = \min\left(T_M, \min_{(a,u)\in\mathcal{L}_m} w_u \log_2(1+\gamma_{a,u})\right) \tag{F.68}$$

(step 9). The goal now is to increase $\{T_{a,u}\}_{(a,u)\in\mathcal{L}_m}$ as much as possible not exceeding T_{M_2}, as long as those involved $a \in \mathcal{A}_g$ can request more resources to increase T^a. Let $\beta \in [0,1]$ be an undetermined parameter, $\{T_{a,u}\}_{(a,u)\in\mathcal{L}_m}$ will be increased by $\beta\cdot(T_{M_2} - T_m)$, i.e., at most, by $T_{M_2} - T_m$ (step 14). Parameter β will be defined later. Let

$$\overline{\mathcal{U}}_a = \{u \in \mathcal{U}_a \mid (a,u) \in \mathcal{L}_m\}, \forall a \in \mathcal{A}_g \tag{F.69}$$

(step 10). Now, we set $T'_{a,u} = T_{a,u} + \beta\cdot(T_{M_2} - T_m)$, $\forall (a,u) \in \mathcal{L}_m$ to increase the involved

throughput rates. Hence, we set $\forall a \in \mathcal{A}_g$

$$\begin{aligned} T^a &= \sum_{u \notin \overline{\mathcal{U}}_a} T_{a,u} + \sum_{u \in \overline{\mathcal{U}}_a} T'_{a,u} = \\ &= \sum_{u \notin \overline{\mathcal{U}}_a} T_{a,u} + \sum_{u \in \overline{\mathcal{U}}_a} (T_{a,u} + \beta \cdot (T_{M_2} - T_m)) = \\ &= \sum_{u \notin \overline{\mathcal{U}}_a} T_{a,u} + \sum_{u \in \overline{\mathcal{U}}_a} T_{a,u} + |\overline{\mathcal{U}}_a| \beta \cdot (T_{M_2} - T_m). \end{aligned} \tag{F.70}$$

Hence, we set $\forall a \in \mathcal{A}_g$

$$\begin{aligned} w^a_{new} &= \frac{T_a}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)} = \frac{\sum\limits_{u \in \mathcal{U}_a} T_{a,u} + |\overline{\mathcal{U}}_a| \beta \cdot (T_{M_2} - T_m)}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)} = \\ &= w^a + \frac{|\overline{\mathcal{U}}_a| \beta \cdot (T_{M_2} - T_m)}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)} \end{aligned} \tag{F.71}$$

(step 12). Now, the aggregation of the new backhaul bandwidth allocation has to be lower than the total bandwidth, i.e.,

$$\begin{aligned} \sum_{a \in \mathcal{A}_g} w^a_{new} &= \sum_{a \in \mathcal{A}_g} \left(w^a + \frac{|\overline{\mathcal{U}}_a| \beta \cdot (T_{M_2} - T_m)}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)} \right) = \\ &= \sum_{a \in \mathcal{A}_g} w^a + \beta \cdot (T_{M_2} - T_m) \sum_{a \in \mathcal{A}_g} \frac{|\overline{\mathcal{U}}_a|}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)} \end{aligned} \tag{F.72}$$

has to be lower or equal than $W_{\mathcal{B}}$. Hence, isolating β we get that necessarily,

$$\beta \leq \frac{W_{\mathcal{B}} - \sum\limits_{a \in \mathcal{A}_g} w^a}{(T_{M_2} - T_m) \sum\limits_{a \in \mathcal{A}_g} \frac{|\overline{\mathcal{U}}_a|}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)}}. \tag{F.73}$$

Hence, we define β as

$$\beta = \min \left(1, \frac{W_{\mathcal{B}} - \sum\limits_{a \in \mathcal{A}_g} w^a}{(T_{M_2} - T_m) \sum\limits_{a \in \mathcal{A}_g} \frac{|\overline{\mathcal{U}}_a|}{\log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)}} \right) \tag{F.74}$$

(step 11). Once the parameter β is derived, we assign $w^a = w^a_{new}$ and $T^a = w^a \log_2\left(1+\gamma^{\mathcal{B}}_{g,a}\right)$, $\forall a \in \mathcal{A}_g$ (step 13). In case that $\beta = 1$ (step 5), we repeat the process defining T_m again and increasing the corresponding throughput rates. This yields the optimal solution.

With limited backbone capacity τ_g. Now, we do not assume that τ_g is unbounded.

Algorithm 13 Optimal solution to CP (6.4). Generic case

1: $w^a \leftarrow W_{\mathcal{B}}^{\min}$, $\forall a \in \mathcal{A}_g$.
2: $T^a \leftarrow w^a \log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)$, $\forall a \in \mathcal{A}_g$.
3: Derive $\{T_u\}_{u\in\mathcal{U}_g}$ and $\{T_{a,u}\}_{u\in\mathcal{U}_a}$, $\forall a \in \mathcal{A}_g$ by solving optimal resource allocation for *gNB* g assuming unbounded τ_g and also $\forall a \in \mathcal{A}_g$ assuming a backhaul limitation of T^a, as detailed in Subsection F.1.
4: $\beta \leftarrow 1$.
5: **while** $\beta = 1$ **do**
6: $T_m \leftarrow \min\{T_{a,u} \,|\, T_{a,u} < w_u \log_2(1+\gamma_{g,a}), a \in \mathcal{A}_g, u \in \mathcal{U}_a\}$.
7: $\mathcal{L}_m \leftarrow \left\{(a,u) \in \mathcal{A}_g \times \bigcup_{a\in\mathcal{A}_g} \mathcal{U}_a \;|\; T_{a,u} = T_m\right\}$.
8: $T_M \leftarrow \min\{T_{a,u} \;|\; T_{a,u} > T_m, a \in \mathcal{A}_g, u \in \mathcal{U}_a\}$.
9: $T_{M_2} \leftarrow \min\left(T_M, \min_{(a,u)\in\mathcal{L}_m} w_u \log_2(1+\gamma_{a,u})\right)$.
10: $\overline{\mathcal{U}}_a \leftarrow \{u \in \mathcal{U}_a \;|\; (a,u) \in \mathcal{L}_m\}$, $\forall a \in \mathcal{A}_g$.
11: $\beta \leftarrow \min\left(1, \frac{W_{\mathcal{B}} - \sum_{a\in\mathcal{A}_g} w^a}{(T_{M_2}-T_m)\cdot \sum_{a\in\mathcal{A}_g} \frac{|\overline{\mathcal{U}}_a|}{\log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)}}\right)$.
12: $w^a \leftarrow w^a + \frac{|\overline{\mathcal{U}}_a|\cdot\beta\cdot\left(T_{M_2}-T_m\right)}{\log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)}$, $\forall a \in \mathcal{A}_g$.
13: $T^a \leftarrow w^a \log_2\left(1+\gamma_{g,a}^{\mathcal{B}}\right)$, $\forall a \in \mathcal{A}_g$.
14: $T_{a,u} \leftarrow T_{a,u} + \beta\cdot(T_{M_2} - T_m)$, $\forall (a,u) \in \mathcal{L}_m$.
15: **end while**
16: $T^a \leftarrow \sum_{u\in\mathcal{U}_a} T_{a,u}$, $\forall a \in \mathcal{A}_g$.
17: **if** $\sum_{u\in\mathcal{U}_g} T_u + \sum_{a\in\mathcal{A}_g} T^a > \tau_g$ **then**
18: Apply Algorithm 10 to $n = |\mathcal{U}_g| + \sum_{a\in\mathcal{A}_g} |\mathcal{U}_a|$, $\{x_i\}_{i=1}^n = \{T_u\}_{u\in\mathcal{U}_g} \cup \bigcup_{a\in\mathcal{A}_g} \{T_{a,u}\}_{u\in\mathcal{U}_a}$, $X = \tau_g$.
19: **end if**

Hence, we now find the optimal solution by solving the problem as above, assuming that τ_g is indeed unbounded, and hence progressively decreasing highest individual throughput rates T_u, $\forall u \in \mathcal{U}_g \cup \bigcup_{a\in\mathcal{A}_g} \mathcal{U}_a$ until $\sum_{u\in\mathcal{U}_g} T_u + \sum_{a\in\mathcal{A}_g} \sum_{u\in\mathcal{U}_a} T_u = \tau_g$, as done also in the analyzed cases in Subsections F.1 and F.2 (step 18 of Algorithm 13).

References

[1] E. Arribas and V. Mancuso, "Multi-Path D2D Leads to Satisfaction," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2017, pp. 1–7.

[2] E. Arribas, V. Mancuso, and V. Cholvi, "Fair Cellular Throughput Optimization with the Aid of Coordinated Drones," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2019, pp. 295–300.

[3] E. Arribas, A. F. Anta, D. Kowalski, V. Mancuso, M. Mosteiro, J. Widmer, and P. W. Wong, "Optimizing mmWave Wireless Backhaul Scheduling," *IEEE Transactions on Mobile Computing*, 2019.

[4] E. Arribas, V. Mancuso, and V. Cholvi, "Coverage Optimization with a Dynamic Network of Drone Relays," *IEEE Transactions on Mobile Computing*, 2019.

[5] E. Arribas and V. Mancuso, "Millimeter-Wave Meets D2D: A Survey," *5G REF Wiley & Sons*, 2020.

[6] ——, "Achieving Per-Flow Satisfaction with Multi-Path D2D," *Ad Hoc Networks*, 2020.

[7] E. Arribas, V. Mancuso, and V. Cholvi, "Fair Throughput Optimization with a Dynamic Network of Drone Relays," *Under revision in Trasactions on Networking*.

[8] W. Mohr, "5G empowering vertical industries," Tech. Rep., 2016.

[9] M. Maternia *et al.*, "5G PPP use cases and performance evaluation models," 5G-PPP, Tech. Rep., Apr. 2016, White Paper.

[10] C. Liu, K. Sundaresan, M. Jiang, S. Rangarajan, and G. Chang, "The case for re-configurable backhaul in cloud-RAN based small cell networks," in *IEEE INFOCOM*, 2013.

[11] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, 2014.

[12] 3GPP, "LTE Device to Device Proximity Services; User Equipment (UE) radio transmission and reception," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.877, 03 2015, version 12.0.0. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2576

[13] *WiFi Direct Alliance*, http://www.wi-fi.org/discover-wi-fi/wi-fi-direct.

[14] A. Asadi and V. Mancuso, "WiFi Direct and LTE D2D in action," in *Wireless Days (WD), 2013 IFIP*. IEEE, 2013, pp. 1–8.

[15] A. Asadi, V. Mancuso, and R. Gupta, "An SDR-based experimental study of outband D2D communications," in *INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, IEEE*. IEEE, 2016, pp. 1–9.

[16] ——, "DORE: an experimental framework to enable outband D2D relay in cellular networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2930–2943, 2017.

[17] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[18] W. Feng, Y. Li, D. Jin, L. Su, and S. Chen, "Millimetre-Wave Backhaul for 5G Networks: Challenges and Solutions," *Sensors*, vol. 16, no. 6, Jun. 2016.

[19] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Communications Magazine*, vol. 54, no. 11, 2016.

[20] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[21] S. Cicalò and V. Tralli, "QoS-aware Admission Control and Resource Allocation for D2D Communications Underlaying Cellular Networks," *IEEE Transactions on Wireless Communications*, 2018.

[22] A. Abrardo, G. Fodor, and B. Tola, "Network coding schemes for D2D communications based relaying for cellular coverage extension," *Transactions on Emerging Telecommunications Technologies*, 2015.

[23] W. Cao, G. Feng, S. Qin, and M. Yan, "Cellular offloading in heterogeneous mobile networks with D2D communication assistance," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4245–4255, 2017.

[24] A. Asadi, V. Mancuso, and P. Jacko, "Floating band D2D: exploring and exploiting the potentials of adaptive D2D-enabled networks," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2015 IEEE 16th International Symposium on a.* IEEE, 2015, pp. 1–9.

[25] 3GPP, "Technical Specification Group Services and System Aspects; Policy and charging control architecture (Release 13)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 23.203, 2015, version 13.4.0.

[26] Y. Li, M. C. Gursoy, S. Velipasalar, and J. Tang, "Joint mode selection and resource allocation for D2D communications via vertex coloring," *arXiv preprint arXiv:1708.00872*, 2017.

[27] Y. Zhang, C.-Y. Wang, and H.-Y. Wei, "Incentive Compatible Overlay D2D System: A Group-Based Framework without CQI Feedback," *IEEE Transactions on Mobile Computing*, 2018.

[28] S. Wen, X. Zhu, X. Zhang, and D. Yang, "QoS-aware mode selection and resource allocation scheme for device-to-device (D2D) communication in cellular networks," in *2013 IEEE International Conference on Communications Workshops (ICC).* IEEE, 2013, pp. 101–105.

[29] S. Maghsudi and D. Niyato, "On transmission mode selection in D2D-enhanced small cell networks," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 618–621, 2017.

[30] F. H. Khan, Y.-J. Choi, and S. Bahk, "Opportunistic mode selection and RB assignment for D2D underlay operation in LTE networks," in *Vehicular Technology Conference (VTC Spring), 2014 IEEE 79th.* IEEE, 2014, pp. 1–5.

[31] D. Della Penda, L. Fu, and M. Johansson, "Mode selection for energy efficient D2D communications in dynamic TDD systems," in *2015 IEEE International Conference on Communications (ICC).* IEEE, 2015, pp. 5404–5409.

[32] E. Datsika, A. Antonopoulos, N. Zorba, and C. Verikoukis, "Cross-network performance analysis of network coding aided cooperative outband D2D communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3176–3188, 2017.

[33] A. Asadi and V. Mancuso, "Network-assisted outband D2D-clustering in 5G cellular networks: theory and practice," *IEEE Transactions on Mobile Computing*, vol. 16, no. 8, pp. 2246–2259, 2017.

[34] J. García-Rois, F. Gómez-Cuba, M. R. Akdeniz, F. J. González-Castaño, J. C. Burguillo, S. Rangan, and B. Lorenzo, "On the analysis of scheduling in dynamic duplex multihop mmWave cellular systems," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6028–6042, 2015.

[35] D. Yuan, H.-Y. Lin, J. Widmer, and M. Hollick, "Optimal joint routing and scheduling in millimeter-wave cellular networks," in *IEEE INFOCOM*, 2018, pp. 1205–1213.

[36] B. Ma, H. Shah-Mansouri, and V. W. Wong, "Full-duplex Relaying for D2D Communication in mmWave based 5G Networks," *IEEE Transactions on Wireless Communications*, 2018.

[37] G. H. Sim, A. Loch, A. Asadi, V. Mancuso, and J. Widmer, "5G millimeter-wave and D2D symbiosis: 60 GHz for proximity-based services," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 140–145, 2017.

[38] S. Wu, R. Atat, N. Mastronarde, and L. Liu, "Coverage analysis of D2D relay-assisted millimeter-wave cellular networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[39] ——, "Improving the coverage and spectral efficiency of millimeter-wave cellular networks using device-to-device relays," *IEEE Transactions on Communications*, vol. 66, no. 5, pp. 2251–2265, 2018.

[40] J. Kim and A. F. Molisch, "Quality-aware millimeter-wave device-to-device multi-hop routing for 5G cellular networks," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 5251–5256.

[41] S. Biswas, S. Vuppala, J. Xue, and T. Ratnarajah, "On the performance of relay aided millimeter wave networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 576–588, 2015.

[42] N. Eshraghi, B. Maham, and V. Shah-Mansouri, "Millimeter-wave device-to-device multi-hop routing for multimedia applications," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[43] N. Wei, X. Lin, and Z. Zhang, "Optimal relay probing in millimeter-wave cellular systems with device-to-device relaying," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10 218–10 222, 2016.

[44] E. Turgut and M. C. Gursoy, "Energy efficiency in relay-assisted mmwave cellular networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2016, pp. 1–5.

[45] X. Lin and J. G. Andrews, "Connectivity of millimeter wave networks with multi-hop relaying," *IEEE Wireless Communications Letters*, vol. 4, no. 2, pp. 209–212, 2015.

[46] B. Xie, Z. Zhang, and R. Q. Hu, "Performance study on relay-assisted millimeter wave cellular networks," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2016, pp. 1–5.

[47] Y. Niu, Y. Liu, Y. Li, Z. Zhong, B. Ai, and P. Hui, "Mobility-Aware Caching Scheduling for Fog Computing in mmWave Band," *IEEE Access*, vol. 6, pp. 69 358–69 370, 2018.

[48] J. Qiao, X. S. Shen, J. W. Mark, Q. Shen, Y. He, and L. Lei, "Enabling device-to-device communications in millimeter-wave 5G cellular networks," *IEEE Communications Magazine*, vol. 53, no. 1, pp. 209–215, 2015.

[49] G. H. Sim, A. Asadi, A. Loch, M. Hollick, and J. Widmer, "Opp-relay: Managing directionality and mobility issues of millimeter-wave via D2D communication," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2017, pp. 144–151.

[50] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 176–189, 2015.

[51] F. H. Kumbhar, N. Saxena, and A. Roy, "Reliable relay: Autonomous social D2D paradigm for 5G LoS communications," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1593–1596, 2017.

[52] A. Orsino, R. Kovalchukov, A. Samuylov, D. Moltchanov, S. Andreev, Y. Koucheryavy, and M. Valkama, "Caching-aided collaborative D2D operation for predictive data dissemination in industrial IoT," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 50–57, 2018.

[53] Y. Ghasempour, C. R. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11 ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.

[54] C. R. da Silva, A. Lomayev, C. Chen, and C. Cordeiro, "Analysis and Simulation of the IEEE 802.11 ay Single-Carrier PHY," in *IEEE ICC*, 2018, pp. 1–6.

[55] Y. Niu, C. Gao, Y. Li, L. Su, D. Jin, and A. V. Vasilakos, "Exploiting device-to-device communications in joint scheduling of access and backhaul for mmWave small cells," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2052–2069, 2015.

[56] V. Genc, S. Murphy, and J. Murphy, "Performance analysis of transparent relays in 802.16j mmr networks," in *2008 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops*, 2008.

[57] W. Guo, C. Devine, and S. Wang, "Performance analysis of micro unmanned airborne communication relays for cellular networks," in *IEEE CSNDSP*. IEEE, 2014.

[58] W. Guo and T. O'Farrell, "Relay deployment in cellular networks: Planning and optimization," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 8, 2013.

[59] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (UAVs): Optimal transport theory for hover time optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8052–8066, 2017.

[60] R. Ferrús, H. Koumaras, O. Sallent, G. Agapiou, T. Rasheed, M.-A. Kourtis, C. Boustie, P. Gélard, and T. Ahmed, "SDN/NFV-enabled satellite communications networks: Opportunities, scenarios and challenges," *Physical Communication*, vol. 18, pp. 95–112, 2016.

[61] S. Katikala, "Google™Project Loon," *InSight: Rivier Academic Journal*, vol. 10, no. 2, pp. 1–6, 2014.

[62] T. Simonite, "Meet Facebook's stratospheric Internet drone," *MIT Technology Review. Retrieved June*, vol. 8, p. 2016, 2015.

[63] C. Vitale, V. Mancuso, and G. Rizzo, "Modelling D2D communications in cellular access networks via Coupled Processors," in *2015 7th International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2015, pp. 1–8.

[64] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *IEEE ICC, 2016*.

[65] I. Strumberger, M. Sarac, D. Markovic, and N. Bacanin, "Moth search algorithm for drone placement problem," *International Journal of Computers*, vol. 3, 2018.

[66] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, 2016.

[67] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[68] A. Hayajneh, S. Zaidi, D. McLernon, and M. Ghogho, "Optimal dimensioning and performance analysis of drone-based wireless communications," in *GC Wkshps*. IEEE, 2016.

[69] W. Wang, H. Dai, C. Dong, X. Cheng, X. Wang, P. Yang, G. Chen, and W. Dou, "Placement of Unmanned Aerial Vehicles for Directional Coverage in 3D Space," *IEEE/ACM Transactions on Networking*, 2020.

[70] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[71] R. Petrolo, Y. Lin, and E. Knightly, "ASTRO: Autonomous, sensing, and tetherless networked drones," in *ACM DroNet, 2018*, pp. 1–6.

[72] L. Wang, B. Hu, and S. Chen, "Energy efficient placement of a drone base station for minimum required transmit power," *IEEE Wireless Communications Letters*, 2018.

[73] W. Mei, Q. Wu, and R. Zhang, "Cellular-connected UAV: Uplink association, power control and interference coordination," *arXiv preprint arXiv:1807.08218*, 2018.

[74] O. Andryeyev and A. Mitschele-Thiel, "Increasing the cellular network capacity using self-organized aerial base stations," in *ACM DroNet 2017*, pp. 37–42.

[75] Y. Zeng, R. Zhang, and T. J. Lim, "Throughput maximization for UAV-enabled mobile relaying systems," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 4983–4996, 2016.

[76] Y. Chen, N. Zhao, Z. Ding, and M.-S. Alouini, "Multiple UAVs as Relays: Multi-Hop Single Link Versus Multiple Dual-Hop Links," *IEEE Transactions on Wireless Communications*, vol. 17, no. 9, pp. 6348–6359, 2018.

[77] G. Zhang, H. Yan, Y. Zeng, M. Cui, and Y. Liu, "Trajectory Optimization and Power Allocation for Multi-Hop UAV Relaying Communications," *IEEE Access*, vol. 6, pp. 48 566–48 576, 2018.

[78] A. Fotouhi, M. Ding, and M. Hassan, "Dynamic base station repositioning to improve spectral efficiency of drone small cells," in *WoWMoM, 2017*. IEEE, pp. 1–9.

[79] P. Casteljau, "Mathématiques et CAO. Vol. 2: formes à pôles," *Hermes*, 1985.

[80] O. Sahingoz, "Generation of bezier curve-based flyable trajectories for multi-UAV systems with parallel genetic algorithm," *Journal of Intelligent & Robotic Systems*, vol. 74, no. 1-2, pp. 499-511, 2014.

[81] M. Moradi, K. Sundaresan, A. Chai, S. Rangarajan, and M. Mao, "SkyCore: Moving Core to the Edge for Untethered and Reliable UAV-based LTE networks," in *ACM MobiCom, 2018*.

[82] R. He, Z. Zhong, B. Ai, L. Xiong, and J. Ding, "The effect of reference distance on path loss prediction based on the measurements in high-speed railway viaduct scenarios," in *Communications and Networking in China (CHINACOM), 2011 6th International ICST Conference on*. IEEE, 2011, pp. 1201–1205.

[83] J. S. Seybold, *Introduction to RF propagation*. John Wiley & Sons, 2005.

[84] J. M. Lucas and M. S. Saccucci, "Exponentially weighted moving average control schemes: properties and enhancements," *Technometrics*, vol. 32, no. 1, pp. 1–12, 1990.

[85] Y. Censor, "Pareto optimality in multiobjective problems," *Applied Mathematics and Optimization*, vol. 4, no. 1, pp. 41–59, 1977.

[86] A. Asadi, "Opportunistic cellular communications with clusters of dual-radio mobiles," Ph.D. dissertation, Universidad Carlos III de Madrid, Spain, 2012.

[87] W. Sun and Y.-X. Yuan, *Optimization theory and methods: nonlinear programming*. Springer Science & Business Media, 2006, vol. 1.

[88] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica: Journal of the Econometric Society*, pp. 497–520, 1960.

[89] M. Belleschi, G. Fodor, and A. Abrardo, "Performance analysis of a distributed resource allocation scheme for D2D communications," in *2011 ieee globecom workshops (gc wkshps)*. IEEE, 2011, pp. 358–362.

[90] R. Liu, G. Yu, F. Qu, and Z. Zhang, "Device-to-device communications in unlicensed spectrum: Mode selection and resource allocation," *IEEE Access*, vol. 4, pp. 4720–4729, 2016.

[91] G. Wunder, M. Kasparick, S. ten Brink, F. Schaich, T. Wild, I. Gaspar, E. Ohlmer, S. Krone, N. Michailow, A. Navarro *et al.*, "5GNOW: Challenging the LTE design paradigms of orthogonality and synchronicity," in *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th.* IEEE, 2013, pp. 1–5.

[92] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[93] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, ser. Lecture Notes in Control and Information Sciences, V. Blondel, S. Boyd, and H. Kimura, Eds. Springer-Verlag Limited, 2008, pp. 95–110, http://stanford.edu/~boyd/graph_dcp.html.

[94] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A Quantitative Measure of Fairness and Discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA, 1984*, 1984.

[95] F. Wang, L. Song, Z. Han, Q. Zhao, and X. Wang, "Joint scheduling and resource allocation for device-to-device underlay communication," in *2013 IEEE wireless communications and networking conference (WCNC).* IEEE, 2013, pp. 134–139.

[96] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of Inter-Satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2442–2473, Fourthquarter 2016.

[97] P. Puri, P. Garg, and M. Aggarwal, "Bi-directional relay-assisted FSO communication systems over strong turbulence channels with pointing errors," *International Journal of Communication Systems*, vol. 30, no. 4.

[98] T. Nitsche, G. Bielsa, I. Tejado, A. Loch, and J. Widmer, "Boon and bane of 60 GHz networks: Practical insights into beamforming, interference, and frame level operation," in *ACM CoNEXT*, 2015.

[99] G. Bielsa, A. Loch, I. Tejado, T. Nitsche, and J. Widmer, "60 GHz Networking: Mobility, Beamforming, and Frame Level Operation From Theory to Practice," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.

[100] M. M. Halldórsson, G. Kortsarz, P. Mitra, and T. Tonoyan, "Spanning Trees With Edge Conflicts and Wireless Connectivity," in *ICALP 2018*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 107, 2018, pp. 158:1–158:15.

[101] S. A. Jafar, "Topological interference management through index coding," *Transactions on Information Theory*, vol. 60, no. 1, pp. 529–568, 2014.

[102] M. Jost, J. S. Gautam, L. G. Gomes, R. Reese, E. Polat, M. Nickel, J. M. Pinheiro, A. L. Serrano, H. Maune, G. P. Rehder *et al.*, "Miniaturized liquid crystal slow wave phase shifter based on nanowire filled membranes," *IEEE Microwave and Wireless Components Letters*, vol. 28, no. 8, pp. 681–683, 2018.

[103] S. F. Jilani, M. O. Munoz, Q. H. Abbasi, and A. Alomainy, "Millimeter-Wave Liquid Crystal Polymer Based Conformal Antenna Array for 5G Applications," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 1, pp. 84–88, 2019.

[104] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS 36.213, V 13.0.0.

[105] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA) Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS 36.321, V 8.0.0.

[106] D. Zuckerman, "Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number," *Theory OF Computing*, vol. 3, pp. 103–128, 2007.

[107] M. Cieliebak, S. Eidenbenz, A. Pagourtzis, and K. Schlude, "On the Complexity of Variations of Equal Sum Subsets," *Nord. J. Comput.*, vol. 14, no. 3, pp. 151–172, 2008.

[108] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, no. 3, pp. 410–421, 1979.

[109] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, no. 4, pp. 373–396, 1984.

[110] C. E. Shannon, "A theorem on coloring the lines of a network," *Journal of Mathematics and Physics*, vol. 28, no. 1, pp. 148–152, 1949.

[111] P. Sanders and D. Steurer, "An asymptotic approximation scheme for multigraph edge coloring," *ACM Transactions on Algorithms (TALG)*, vol. 4, no. 2, p. 21, 2008.

[112] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band," *IEEE Std 802.11ad-2012*.

[113] M. W. Rousstia, "Switched-beam antenna array design for millimeter-wave applications," *PDEng Report, SAI-ICT, Eindhoven University of Technology*, 2011.

[114] D. Steinmetzer, D. Wegemer, M. Schulz, J. Widmer, and M. Hollick, "Compressive Millimeter-Wave Sector Selection in Off-the-Shelf IEEE 802.1 lad Devices," in *Proceedings of CoNEXT, 2017*.

[115] D. Steinmetzer, D. Wegemer, and M. Hollick. (2018) Talon Tools: The Framework for Practical IEEE 802.11ad Research. [Online]. Available: https://seemoo.de/talon-tools/

[116] S. K. Saha, H. Assasa, A. Loch, N. M. Prakash, R. Shyamsunder, S. Aggarwal, D. Steinmetzer, D. Koutsonikolas, J. Widmer, and M. Hollick, "Fast and infuriating: Performance and pitfalls of 60 ghz wlans based on consumer-grade hardware," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 2018, pp. 1–9.

[117] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.

[118] S. Boettcher and A. Percus, "Optimization with extremal dynamics," *Phys. Rev. Lett.*, vol. 86, pp. 5211–5214, 2001.

[119] R. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *ICC*. IEEE, 2016.

[120] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-Connected UAV: Potentials, Challenges and Promising Technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2019.

[121] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *IEEE GLOBECOM, 2014*, pp. 2898–2904.

[122] ITU-R, "Propagation data and prediction methods required for the design of terrestrial line-of-sight systems," 2015.

[123] A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A survey of channel modeling for UAV communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804-2821, 2018.

[124] J. S. Seybold, *Introduction to RF propagation*. John Wiley & Sons, 2005.

[125] G. Das, R. Fraser, A. Lóopez-Ortiz, and B. Nickerson, "On the discrete unit disk cover problem," *International Journal of Computational Geometry & Applications*, vol. 22, no. 05, pp. 407–419, 2012.

[126] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A los map approach," in *IEEE ICC, 2017*.

[127] J. Munkres, "Algorithms for the assignment and transportation problems," *Journal of the society for industrial and applied mathematics*, vol. 5, no. 1, pp. 32–38, 1957.

[128] L. Zanzi, F. Giust, and V. Sciancalepore, "M^{2}EC: A multi-tenant resource orchestration in multi-access edge computing systems," in *IEEE WCNC, 2018*, April 2018, pp. 1–6.

[129] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, Fourthquarter 2016.

[130] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, 2015.

[131] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[132] B. V d Bergh, A. Chiumento, and S. Pollin, "LTE in the sky: trading off propagation benefits with interference costs for aerial nodes," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 44–50, 2016.

[133] J. Lyu, Y. Zeng, and R. Zhang, "Cyclical multiple access in UAV-aided communications: A throughput-delay tradeoff," *IEEE Wireless Communications Letters*, vol. 5, no. 6, pp. 600–603, 2016.

[134] W. Pan and G. Cheng, "QoE assessment of encrypted YouTube adaptive streaming for energy saving in Smart Cities," *IEEE Access*, vol. 6, pp. 25 142–25 156, 2018.

[135] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Transactions on circuits and systems for video technology*, vol. 11, no. 3, pp. 301–317, 2001.

[136] A. Asadi and V. Mancuso, "On the compound impact of opportunistic scheduling and D2D communications in cellular networks," in *ACM MSWiM, 2013*, pp. 279–288.

[137] 3GPP TR38.901, "Study on channel model for frequencies from 0.5 to 100 GHz, V14.0.0."

[138] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile computing*. Springer, 1996, pp. 153-181.

[139] S. Mao, "Fundamentals of communication networks," in *Cognitive Radio Communications and Networks*. Elsevier, 2010, pp. 201–234.

[140] E. Kalantari, I. Bor-Yaliniz, A. Yongacoglu, and H. Yanikomeroglu, "User association and bandwidth allocation for terrestrial and aerial base stations with backhaul considerations," in *IEEE PIMRC,2017*, pp. 1–6.

[141] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Transactions on Wireless Communications*, vol. 16, no. 11, pp. 7574–7589, 2017.

[142] S. Rohde and C. Wietfeld, "Interference Aware Positioning of Aerial Relays for Cell Overload and Outage Compensation," in *IEEE VTC, 2012*.

[143] F. Giust, V. Sciancalepore, D. Sabella, M. C. Filippou, S. Mangiante, W. Featherstone, and D. Munaretto, "Multi-access edge computing: The driver behind the wheel of 5G-connected cars," *IEEE Communications Standards Magazine*, vol. 2, no. 3, pp. 66–73, 2018.

[144] L. Daniel and K. Narayanan, "Congestion control 2: Utility, fairness, and optimization in resource allocation," *Mathematical Modelling for Computer Networks-Part I*, pp. 2–1, 2013.

[145] L. Buzna and R. Carvalho, "Controlling congestion on complex networks: fairness, efficiency and network structure," *Scientific reports*, vol. 7, no. 1, p. 9152, 2017.

[146] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[147] S. Burer and A. N. Letchford, "Non-convex mixed-integer nonlinear programming: A survey," *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012.

[148] M. Chiang *et al.*, "Geometric programming for communication systems," *Foundations and Trends® in Communications and Information Theory*, vol. 2, no. 1–2, pp. 1–154, 2005.

[149] M. L. Fisher, R. Jaikumar, and L. N. Van Wassenhove, "A multiplier adjustment method for the generalized assignment problem," *Management Science*, vol. 32, no. 9, pp. 1095–1103, 1986.

[150] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162–166, 2006.

[151] D. G. Cattrysse and L. N. Van Wassenhove, "A survey of algorithms for the generalized assignment problem," *European journal of operational research*, vol. 60, no. 3, pp. 260–272, 1992.

[152] A. Fréville, "The multidimensional 0–1 knapsack problem: An overview," *European Journal of Operational Research*, vol. 155, no. 1, pp. 1–21, 2004.

[153] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," *European Journal of Operational Research*, vol. 123, no. 2, pp. 394–407, 2000.

[154] Y. Nesterov and A. Nemirovskii, *Interior-point polynomial algorithms in convex programming*. Siam, 1994, vol. 13.

[155] S. Bubeck *et al.*, "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[156] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 247–258.

[157] G. Last and M. Penrose, *Lectures on the Poisson process*. Cambridge University Press, 2017, vol. 7.

[158] G. Miao, J. Zander, K. W. Sung, and S. B. Slimane, *Fundamentals of mobile data networks*. Cambridge University Press, 2016.

[159] S. Mosleh, L. Liu, and J. Zhang, "Proportional-fair resource allocation for coordinated multi-point transmission in LTE-advanced," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5355–5367, 2016.

[160] R. Li and P. Patras, "Max-Min Fair Resource Allocation in Millimetre-Wave Backhauls," *IEEE Transactions on Mobile Computing*, 2019.

[161] J. del Peral-Rosado *et al.*, "Survey of cellular mobile radio localization methods: from 1G to 5G," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1124–1148, 2017.

[162] C. Scheideler, A. W. Richa, and P. Santi, "An $O(\log n)$ dominating set protocol for wireless ad-hoc networks under the physical interference model," in *ACM MobiHoc*. ACM, 2008, pp. 91–100.

[163] T. Kesselheim, "Dynamic packet scheduling in wireless networks," in *Proceedings of ACM PODC, 2012*, pp. 281–290.

[164] D. R. Kowalski, H. Kudaravalli, and M. A. Mosteiro, "Ad-hoc Affectance-selective Families for Layer Dissemination," *CoRR*, vol. abs/1703.01704, 2017.

[165] D. R. Kowalski, M. A. Mosteiro, and K. Zaki, "Dynamic Multiple-Message Broadcast: Bounding Throughput in the Affectance Model," *CoRR*, vol. abs/1512.00540, 2015.

[166] T. F. Hain, A. L. Ahmad, S. V. R. Racherla, and D. D. Langan, "Fast, precise flattening of cubic Bezier path and offset curves," *Computers & Graphics*, vol. 29, no. 5, pp. 656–666, 2005.